

EPODOC / EPO

PN - JP11282686 A 19991015  
 PD - 1999-10-15  
 PR - JP19980083243 19980330  
 OPD - 1998-03-30  
 TI - NETWORK COMPUTER SYSTEM  
 IN - IGARASHI RIKA; KAWABATA RITSUKO; SATOYAMA MOTOAKI  
 PA - HITACHI LTD  
 IC - G06F9/445 ; G06F13/00 ; G06F15/16

© WPI / DERWENT

TI - Program forwarding control system in computer network - forwards suitable application program to client from server after receiving its load demand and judging its system capability, memory capacity and other hardware configuration, using machine component information

PR - JP19980083243 19980330

PN - JP11282686 A 19991015 DW200001 G06F9/445 019pp

PA - (HITA ) HITACHI LTD

IC - G06F9/445 ;G06F13/00 ;G06F15/16

AB - JP11282686 NOVELTY - Client ( 110) outputs a load demand to server ( 120) to require forwarding of a program. A machine component information shows empty memory and system capability of client. Then the server accordingly chooses optimum program version suitable for that client and forwards it to client automatically.

- USE - For forwarding suitable application program to client, from server, in computer network.

- ADVANTAGE - Even if the user does not know about the hardware requirement and application etc, suitable program is forwarded reliably. Hence wastage of time in loading is avoided. DESCRIPTION OF DRAWING(S) - The figure shows the explanatory view of network manufacturing procedure. ( 110) Client; ( 120) Server.

- (Dwg. 1/21)

OPD - 1998-03-30

AN - 2000-004109 [01]

© PAJ / JPO

PN - JP11282686 A 19991015  
 PD - 1999-10-15  
 AP - JP19980083243 19980330  
 IN - KAWABATA RITSUKO; SATOYAMA MOTOAKI; IGARASHI RIKA  
 PA - HITACHI LTD  
 TI - NETWORK COMPUTER SYSTEM

AB - PROBLEM TO BE SOLVED: To provide technology for executing an application to be loaded and executed in a comfortable operation environment on a client terminal without having a user be conscious of the constitution of hardware or the operating condition of the application.

- SOLUTION: A program server device 120 receiving a program loading request from a client terminal 110 automatically selects an optimum program capable of realizing an executable operation environment required by a user from execution time priority information set up in each user about the machine constitution information of the terminal 110 and elements for determining a program execution environment and transfers the selected program.

I - G06F9/445 ;G06F13/00 ;G06F13/00 ;G06F15/16

**THIS PAGE BLANK (USPTO)**

[Claim(s)]

[Claim 1] Server equipment equipped with a central processing unit, main storage, external storage, and network communication equipment. A central processing unit, main storage, network communication equipment, the input unit like a keyboard, and the output unit like a display. To the external storage of the program server equipment which is the network computer system equipped with the above, and stores the program which the above-mentioned client terminal unit loads. The application execution condition which defined the power which the client terminal unit which performs the stored application program fills, the "primary control program", the "amount of memory requirements", and the "performance", The machine configuration information which shows the "primary control program", the "amount of empty memory", and the "performance" which the above-mentioned client terminal unit in a connection state has, The execution-time priority information set up for every user about three elements of the "performance", the "amount of empty memory", and the "version" which determine program execution environment, The execution condition of \*\*\*\*\* and the above-mentioned application program, and the above-mentioned machine configuration information, It is characterized by providing a means to choose an execute permission and the optimal program which realizes the execution environment which a user wishes at the time of execution from the priority information at the time of the above-mentioned application execution set up for every user with a client terminal unit, to the above-mentioned program server equipment.

[Claim 2] In an execute permission according to claim 1 and a means to choose the optimal program. If the primary control program under execution is changed on a client terminal unit. When the above-mentioned program server equipment can transmit the program which can be performed to a client terminal unit. From the execution condition and the above-mentioned machine configuration information of the above-mentioned application program. The combination of the primary control program which can be performed, and an application program is extracted. Whether the execution environment of application is prepared by changing the composition of a client terminal unit in use. The network computer system according to claim 1 characterized by performing environmental maintenance for performing an application program automatically by disposal which re-transmits a primary control program only when the means which a user can specify at the time of use is established and a user specifies.

[Claim 3] The network computer system according to claim 1 characterized by choosing the program which realizes the execution environment which uses a different judgment formula and a user wishes with the element with which the user set up the priority highly most in the execution-time priority information which has three elements according to claim 1 which determine the execution environment of application for every user in an execute permission according to claim 1 and a means to choose the optimal program automatically.

[Claim 4] The network computer system according to claim 1 characterized by having a means by which a user or a system administrator can set up beforehand the execution-time priority which has three elements according to claim 1 which determine the execution environment of an application program in application execution-time priority information according to claim 1.

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] Both a client terminal unit and server equipment are connected in a network, and this invention is applied to the network computer system which carries out load and go of the required program from the above-mentioned server equipment to the above-mentioned client terminal unit via a network, and relates to effective technology.

[0002]

[Description of the Prior Art] In the conventional network computer system, it connects with server equipment at during starting, and a client terminal unit carries out load and go of the primary control program as stated to "the Nikkei electronics 1997.4.21 (no.687)" from server equipment. When performing an application program with a client terminal unit, the name and version of an application program which a user wants to perform are specified, and a load demand is published to server equipment. Server equipment searches the application program which had the demand from the client terminal unit from the application program group in which it is stored by external storage, and has the function to transmit to a client terminal unit.

[0003]

[Problem(s) to be Solved by the Invention] In the aforementioned conventional network computer system

Various versions which have an operating condition in the application program stored in server equipment separately are intermingled, and whether it is an execute permission In order to be dependent on adjustment with the configuration information (hardware and primary control program under execution) of a client terminal unit which published the load demand and to load the program which can be performed In advance, the user needed to investigate the adjustment of the machine configuration information of a client terminal unit, and the operating condition of the application program which it is going to perform, and it needed to judge which program (version) is an execute permission.

[0004] Moreover, when the program which can be performed with the composition of a client terminal unit in use does not exist as a result of the judgment of the above-mentioned adjustment (for example, when only a primary control program is mismatching), in order to perform an application program, the user itself had to devise investigation of a factor and the solution which demanded and which cannot perform application of improving environment, and environmental maintenance had to perform by the handicraft by reloading the demanded application and an adjustable primary control program.

[0005] Moreover, since it did not become clear whether the program which published the load demand is an execute permission unless load and go is carried out to a client terminal unit at once when the user does not fully grasp information required for the judgment of the above-mentioned adjustment, the user might repeat trial and error and might waste time on loading of the program which cannot be performed. Moreover, when the program which can be performed in server equipment does not exist, the user has not recognized that the program which can be performed in server equipment does not exist until it carried out load and go of all the programs and ended in failure.

[0006] Moreover, in the load and go of this application program causing the shortage of memory of a client terminal unit, and degradation of a performance, when the loaded application program can be performed, it cannot be called an operating environment comfortable for a user. A comfortable operating environment may show the environment where the newest version of an application program can be performed, when how to catch by the user differs and the goodness of a performance is shown, and when the environment where two or more applications can be performed simultaneously is shown. There was no guarantee which can be offered in the conventional network computer system by the operating environment which program execution environment is inevitably determined and a user senses is comfortable from the machine configuration information of a client terminal and the program execution conditions which carried out load and go to the beginning. Moreover, it did not have a means by which a user set up program execution environment, either.

[0007] The server equipment which the purpose of this invention solved the trouble of the conventional technology mentioned above, and received the load demand from the client terminal unit [ when the program which can be performed is chosen, it is the application which the user demanded and the program which can be performed exists in server equipment before a transfer of a program ] It is in offering the network computer system which can carry out load and go of the demanded program by once certainly, without a user being conscious of machine configuration information or the operating condition of a program.

[0008] If a primary control program is changed on a client terminal unit with the load demand, the purpose of this invention When the application program which can perform server equipment can be transmitted Whether the execution environment of application is prepared by changing the composition of a client terminal unit By disposal which re-transmits the program demanded only when the means which a user can specify at the time of use was established and a user specified, and a primary control program with adjustment It is in offering the network computer system which can obtain the execution environment of application which the user wished.

[0009] It is the application which the purpose of this invention displayed the error message, without transmitting the demanded program when the program which can be performed on a client terminal unit does not exist, even if it re-transmits a primary control program, and the user demanded in server equipment, and is in offering the network computer system which notifies a user of the program which can be performed on a client terminal unit in use not existing.

[0010] The purpose of this invention is the application in which the user did the load demand, and when two or more programs which can be performed exist, it is to offer the network computer system which chooses the optimal program which realizes the operating environment which a user wishes at the time of execution, and it not only can perform, but provides a user with a comfortable operating environment at

it. Moreover, it is in offering the network computer system with which it can have a means by which a user or a system administrator can set up flexibly the execution-time priority according to the operating condition, and a user can tune up an execution environment about three elements of the "performance", the "amount of empty memory", and the "version" which determine the execution environment of an application program.

[0011]

[Means for Solving the Problem] The during-starting connecting means which the aforementioned purpose connects with a client terminal unit with program server equipment after starting according to this invention, and carry out load and go of the primary control program, A network program load means to publish a program transfer request to program server equipment, and to receive a program, A notice means of machine configuration information to notify renewal of a value to the program server equipment in a connection state when change of hardware composition arises with a client terminal unit, A server basic control means like an operating system to perform fundamental control of hardware to \*\*\*\*\* and program server equipment, A program transfer means to transmit a program according to the program load demand from a client terminal unit, The network computer primary control program transmitted according to the demand of a client terminal unit, and an application program, The application execution condition pattern table in which the operating condition of this application program was described, The machine configuration information management data which manages the hardware configuration information of the client terminal unit in a connection state, From this application execution condition pattern table and this machine configuration information It is the application which the user demanded, and the program which can be performed is extracted out of two or more programs which exist in server equipment, and it is attained by having a transfer application selection means to transmit to a client terminal unit.

[0012] This application execution condition pattern table has three, a "primary control program", the "amount of memory requirements", and a "performance", as an item required of the client terminal unit which performs this application, and an indispensable condition (conditions which must be fulfilled in order to perform), and recommendation conditions (conditions with desirable although it is an execute permission filling) are set up only about the value of the item of the "amount of memory requirements", and a "performance." This machine configuration information management data has three items, a "primary control program", the "amount of empty memory", and a "performance", about the client terminal unit in use.

[0013] Moreover, the aforementioned purpose is set in the middle of processing of the aforementioned transfer application selection means. If the primary control program under execution is changed on a client terminal unit When the application program which can perform server equipment can be transmitted The hardware configuration information of a client terminal unit which published the load demand is investigated. The combination of the primary control program which can be performed, and an application program with the demand is searched. By displaying the message which stimulates reloading of the application required of the client terminal unit, and a primary control program with adjustment Only when a means by which a user can specify whether the configuration information of a client is changed and program execution environment is prepared at the time of use is established and a user wishes, a primary control program, When an application program with the demand is transmitted to a client terminal unit and a user does not specify it It is attained, when an error message is displayed and the aforementioned transfer application selection means contains the step which shows a user that the program which had the load demand with machine composition in use cannot be performed, without transmitting a program.

[0014] Moreover, the aforementioned purpose is set in the middle of processing of the aforementioned transfer application selection means. The value of the indispensable condition of the item "a performance" of this application execution condition pattern table that it is going to perform, and an item "the amount of memory requirements" When the value of the item "a performance" of the machine configuration information of the client terminal unit which the user who extracted from this machine configuration information management data is using, and an item "the amount of empty memory" is not filling It is attained when the aforementioned transfer application selection means contains the step which shows that the program which can be performed in server equipment does not exist to a client terminal unit and which displays an error message.

[0015] Moreover, the "performance" to which the aforementioned purpose determines the execution environment of application as program server equipment, It has application execution-environment tuning data in which the execution-time priority of each element of "the amount of empty memory" and a "version" is shown for every user and every session. This application execution condition pattern table, These application execution-environment tuning data are referred to to the program which was extracted from this machine configuration management information and which can be performed. The judgment formula suitable for the highest element of the priority which the user set up is chosen and performed, and it is attained when the aforementioned transfer application selection means is equipped with the step which chooses the optimal application program which provides a user with a comfortable execution environment. Moreover, it is attained by equipping program server equipment with a means by which a user or a system administrator sets up the execution-time priority of the three aforementioned elements which determine the execution environment of application.

[0016]

[Embodiments of the Invention] Below, a drawing explains the network computer management method of this invention, and 1 operation gestalt of equipment in detail.

[0017] Drawing 1 is drawing showing the outline composition of the operation equipment which enforces the network computer management method of this operation gestalt. In drawing 1, 110 is a client terminal unit and 120 is program server equipment. The client terminal unit 110 and program server equipment 120 are connected in the network. The client terminal unit 110 has a central processing unit 111, the input unit 112 like a keyboard, the output unit 113 like a display, a communication device 114, a nonvolatile storage 115, and main storage 116. In the nonvolatile storage 115, it has the starting connection program 700, the network program load program 800, and the notice program 900 of machine configuration information.

[0018] Program server equipment 120 has a central processing unit 121, a communication device 122, main storage 123, and external storage 124. A primary control program 125, the program transport agent 1000, and the transfer application selection program 1100 are read into main storage 123 from external storage 124, and are performed with a central processing unit 121. External storage 124 stores the network computer primary control program 126, an application program 127, the application execution condition pattern table 200, the machine configuration information management data 300, and the application execution-environment tuning data 400.

[0019] The network computer primary control program 126 stored in the external storage 124 which program server equipment 120 has, and an application program 127 are read into the main storage 116 which the client terminal unit 110 has by the network computer management method of this operation gestalt, and are performed by the central processing unit 111.

[0020] Drawing 2 is drawing which is stored in the external storage 124 of the operation equipment which enforces the network computer management method of this operation gestalt and in which showing an example of the application execution condition pattern table 200. The application execution condition pattern table 200 defines the operating environment needed in case the application program 127 stored in the same external storage 124 is performed in the client terminal unit 110, and is prepared one for every application program.

[0021] Drawing 2 defines the operating environment which the client terminal unit 110 must fill, when performing an application program A in the client terminal unit 110. In drawing 2, 201 is a pattern-recognition number (it is described as "Pattern ID" below) for managing the data of a table for every train. 202 shows the name of the program which can perform an application program A, i.e., the existing name of the version of an application program A. It is shown that a version is new as it is arranged by the old order of a version and a number becomes large sequentially from No. 1 of "Pattern ID."

[0022] 203-205 show the conditions which the hardware composition of the client terminal unit 110 must fulfill, when it is going to perform an application program A. When 203 performs an application program A, performing on the client terminal unit 110 shows the name and version of a primary control program which are demanded. The performance of the power central processing unit 111 in which, as for 205, the client terminal unit 110 fills the amount of memory to which 204 is needed in the hardware composition of the client terminal unit 110 at the time of execution of an application program A, i.e., the value of CPU, is shown.

[0023] An indispensable condition and recommendation conditions exist about 204, the amount of

memory shown in 205, and the value of CPU. Indispensable conditions are conditions which the client terminal unit 110 must fulfill, in order to perform the version of the corresponding application program. Filling are desirable conditions and recommendation conditions are conditions for performing a program more comfortably in the case where it performs in parallel with other applications, or a performance side. [0024] Drawing 3 is drawing showing an example of the machine configuration information management data 300 of operation equipment which enforces the network computer management method of this operation gestalt. The machine configuration information management data 300 stores the hardware configuration information of the client terminal unit 110 linked to program server equipment 120. In drawing 3, the identification number of the client terminal unit 110 which 301 connected with program server equipment 120, and 302-304 show the hardware configuration information of the client terminal unit discriminated by 301. 302 shows the name and version of a primary control program under execution. 303 shows the amount of empty memory. 304 shows the performance of a central processing unit 111, i.e., the carried value of CPU.

[0025] Drawing 4 is drawing showing an example of the application execution-environment tuning data 400 of operation equipment which enforces the network computer management method of this operation gestalt. The application execution-environment tuning data 400 store the execution-time priority information on three attributes, a "performance", the "amount of empty memory", and a "version", that the execution environment of the application program in the client terminal unit 110 is determined, for every user and every session.

[0026] In drawing 4, 401 shows the user-identification number which is using the client terminal unit 110 linked to program server equipment 120, or a session number. 402 has three attributes which show the value of the tuning parameter which determines the execution environment of an application program, and are shown in 403-405.

[0027] The priority of the amount of empty memory of the client terminal unit 110 with which the priority of the performance of the client terminal unit 110 in which 403 performs an application program, and 404 perform an application program, and 405 are the priorities of the newness of the version of the application program performed in the client terminal unit 110. In 402, the sum of three values is set up so that it may surely be set to 1, and an attribute with a larger value means that the priority at the time of execution is high.

[0028] Drawing 5 is drawing showing an example of the setting screen of the application execution-environment tuning data 400 shown in drawing 4. The setting candidate of the application execution-environment tuning data 400 is each user who uses the client terminal unit 110, or a system administrator who has managed the session.

[0029] The setting bar shown in 501 is prepared about each attribute of 403-405, and a user or a system administrator sets up the priority of each attribute by moving the pointer shown in 502. The position of the pointer which the user set up is set to 1 the case where priority is given most, in processing, the case where priority is not given most is set to 0, by where it is placed, it is calculated by the value of 0-1 and the application execution-environment tuning data 400 of aforementioned drawing 4 are filled in. In two or more attributes, it is impossible to set a pointer as the same position, i.e., to set a priority as the same value, and the sum of the value of three attributes is set up so that it may surely be set to 1.

[0030] This program is stored as one of the application programs in program server equipment, and it is contained in the during starting of a network computer at the basic application program group which is transmitted to a client terminal unit simultaneously with a primary control program and which is usually called desktop program, and a setup is performed and a user can make a change, while using a client terminal unit.

[0031] Drawing 6 is drawing showing the data structure of the demand which the client terminal unit 110 publishes to program server equipment 120. The identification number of the client terminal unit 110 which connected 601 to program server equipment 120, and 602 show the instruction which the client terminal unit 110 publishes to program server equipment. The information included according to the kind of instruction which 603 shows supplementary information and is shown in 602 differs. Three kinds shown in (A) - (C) below, "an initial-program transfer" (602-a), "an application transfer" (602-b), and "renewal of machine configuration information" (602-c) exist in the demand which the client terminal unit 110 publishes to program server equipment 120.

[0032] (A) shows the data structure of the demand which the client terminal unit 110 publishes to



program server equipment 120, when an instruction is "an initial-program transfer." The client terminal unit 110 assembles the demand with the data structure of (A) at the same time the power supply was turned on. User information and the hardware configuration information of the client terminal unit 110 are set to the supplementary information 603.

[0033] As for a user-identification child and 603-b, 603-a shows the amount of empty memory of the client terminal unit 110, and the name and version of the network computer primary control program 126 with which 603-d is first loaded to the client terminal unit 110, as for the performance of the central processing unit 111 of the client terminal unit 110, and 603-c. The supplementary information 603 sent to during starting is written in the machine configuration information managed table 300 of aforementioned drawing 3 .

[0034] (B) shows the data structure of the demand which the client terminal unit 110 publishes to program server equipment 120, when an instruction is "an application transfer." A power supply is turned on, and when the client terminal unit 110 in the state where the primary control program 126 was loaded wants to perform the application program 127 stored in the external storage 124 of program server equipment 120, the demand with the data structure of (B) is assembled. Supplementary information 603-e shows the name of an application program to load.

[0035] (C) shows the data structure of the demand which the client terminal unit 110 publishes to program server equipment 120, when an instruction is "renewal of machine configuration information." A power supply is turned on and a primary control program and an application program are loaded, and when change arises in hardware configuration information by extension of memory, or reloading of a primary control program in the client terminal unit 110 in the state of performing, it is the demand which the client terminal unit 110 publishes. Supplementary information 603-f shows the attribute of hardware configuration information with change, and the value after change. The corresponding information in the machine configuration information management data 300 shown in aforementioned drawing 3 is updated by issue of this demand.

[0036] Drawing 7 is a flow chart explaining the procedure of the starting connection program 700 stored in the nonvolatile storage 115 of the client terminal unit 110. the starting connection program 700 -- (1) -- the machine configuration information of the client terminal unit 110 containing the name and version of a primary control program which are loaded to the network address stored beforehand, a user-identification child, the performance (value of CPU) of a central processing unit 111, the amount of empty memory, and the beginning is acquired (Step 701)

[0037] (2) The network program load program 800 is called, assemble a "initial-program transfer request" from the information acquired at Step 701, publish a demand to program server equipment 120, and load the network computer primary control program 127 (Step 800, Steps 801-805 of the after-mentioned [ a detail ]).

[0038] (3) Perform the loaded program (Step 702).

[0039] Drawing 8 is a flow chart explaining the procedure of the network program load program 800 stored in the nonvolatile storage 115 of the client terminal unit 110. The network program load program 800 assembles the program transfer request shown in (1) aforementioned drawing 6 (Step 801).

[0040] (2) Publish the program transfer request assembled in Step 801 to program server equipment 120 (Step 802).

[0041] (3) Connect with program server equipment 120 using a communication device 114 (Step 803).

[0042] (4) Read into main storage 116 the program included in a program transfer request from the external storage 124 which program server equipment 120 has (Step 804).

[0043] (5) Write the read program data in main storage (Step 805).

[0044] Drawing 9 is a flow chart explaining the procedure of the notice program 900 of machine configuration information stored in the nonvolatile storage 115 of the client terminal unit 110. The notice program 900 of machine configuration information acquires the machine configuration information of (1) client terminal unit 110 (Step 901). Machine configuration information has three kinds of "primary control programs" which shows the name and version of "CPU" which shows the performance of a central processing unit, the "amount of empty memory" which shows the amount of empty memory, and the primary control program loaded to the beginning.

[0045] (2) Investigate whether change arose in "CPU" of the client terminal unit 110 (Step 902).

[0046] (3) Investigate whether change arose in the "amount of empty memory" of the client terminal unit



110 (Step 903).

[0047] (4) Investigate whether change arose in the "primary control program" of the client terminal unit 110 (Step 904).

[0048] (5) When change arises in Steps 902-904, assemble "a renewal demand of machine configuration information" from the value after an attribute with change, and change (Step 905).

[0049] (6) When change arises in Steps 902-904, publish "a renewal demand of machine configuration information" assembled in Step 905 to program server equipment 120 (Step 906).

[0050] (7) When change does not arise in Steps 902-904, end processing.

[0051] Drawing 10 is a flow chart explaining the procedure of the program transport agent 1000 stored in the main storage 123 of program server equipment 120. The program transport agent 1000 processes the demand transmitted through a network from the network load program 800 which the client terminal unit 110 has. The program transport agent 1000 receives the demand published from (1) client terminal unit 110 using a communication device 122 (Step 1001).

[0052] (2) Investigate an instruction of the demand which received in Step 1001 (Step 1002).

[0053] (3) When an instruction of the demand which received in Step 1001 is "renewal of machine configuration information", extract the machine configuration information which corresponds in the machine configuration information management data 300 from the network address included in a demand, read the supplementary information included in a demand, and update the value of an attribute with change (Step 1003).

[0054] (4) When an instruction of the demand which received in Step 1001 is "a program transfer", extract information more nearly required than the machine configuration information management data 300, the application execution condition pattern table 200, and the application execution-environment tuning data 400, and choose the optimal application program which provides an execute permission and a user with a comfortable operating environment and which should be transmitted from the network address included in a demand, an application program name, and a user-identification child (Step 1100, Steps 1101-1107 of the after-mentioned [ a detail ]).

[0055] (5) When an instruction of the demand which received in Step 1001 is "a program transfer", transmit the application program chosen in Step 1100 (Step 1005).

[0056] (6) When an instruction of the demand which received in Step 1001 is "an initial-program transfer", read the network address 601 and the supplementary information 603 which are included in a demand, and create machine configuration information (Step 1006).

[0057] (7) When an instruction of the demand which received in Step 1001 is "an initial-program transfer", add the machine configuration information created in 1006 to the machine configuration information management data 300 (Step 1007).

[0058] (8) When an instruction of the demand which received in Step 1001 is "an initial-program transfer", search the network computer primary control program 126 with the load demand contained in the supplementary information 603 on a demand from external storage 124, and transmit to the client terminal unit 110 (Step 1008).

[0059] Drawing 11 is a flow chart explaining the procedure of the transfer application selection program 1100 stored in the main storage 123 of program server equipment 120. The transfer application selection program 1100 opens (1) machine configuration information management data 300 (Step 1101).

[0060] (2) Extract the machine configuration information which searches the machine configuration information management data 300 to a key, and corresponds to it the network address 601 included in the demand published from the client terminal unit 110 (Step 1102).

[0061] (3) Search and open the application execution condition pattern table 200 which corresponds name 603-d of the application program contained in the demand published from the client terminal unit 110 to a key (Step 1103).

[0062] (4) Collate the operating condition contained in the application execution condition pattern table opened in Step 1103 with the machine configuration information of the client terminal unit 110 extracted in Step 1102, and extract the application program which can be performed on the client terminal unit 110 which published the transfer request (Step 1200, Steps 1201-1208 of the after-mentioned [ a detail ]).

[0063] (5) Investigate whether as a result of processing Step 1200, the application program which can be performed exists, or when it investigated and (Step 1104) exists, more than one exist further (Step 1105).

[0064] (6) If even the network computer primary control program under execution is replaced on the

client terminal unit 110 when the application program which can be performed does not exist in Step 1104, as a result of processing Step 1200, it investigates whether the application program which can operate exists, and only when it exists and a user wishes, "primary-control-program resending processing" which resends a primary control program will be performed (Step 2000, Steps 2001-2008 of the after-mentioned [ a detail ]).

[0065] (7) In Step 1105, when two or more application programs which can be performed do not exist, determine the program which was extracted in Step 1200 and which can be performed as a transport agent (Step 1106).

[0066] (8) In Step 1105, give the execution-time priority inspection processing 1400 which chooses the program which offers a comfortable operating environment from the program in which extracted execution is possible in Step 1200 to the user who performed a load demand with reference to the tuning data which search the application execution-environment tuning data 400 to a key, and correspond a user-identification child or a session number when two or more application programs which can be performed exist (1401-1410 of the after-mentioned [ Step 1400 and a detail ]).

[0067] (9) Perform transport-agent terminal-decision processing 1800 of determining the program which should transmit when two or more programs which realize the operating environment which a user wishes further exist after performing execution-time priority inspection processing 1400 in Step 1105, when two or more application programs which can be performed exist, or when the program with which a user's hope fills in all attributes does not exist (Step 1800, Steps 1801-1808 of the after-mentioned [ a detail ]).

[0068] Drawing 12 is a flow chart explaining the procedure of processing 1200 which extracts the program which can be performed, after performing Step 1103 of aforementioned drawing 11 . The processing 1200 which extracts the program which can be performed sets initial value 0 as the variable count showing the number of the program in which (1) execution is possible (Step 1201).

[0069] (2) Investigate whether inspection of the extracted machine configuration information was ended about all the "patterns ID" of an application execution condition pattern table (Step 1202).

[0070] (3) In Step 1202, when inspection of the machine configuration information extracted about all the "patterns ID" of an application execution condition pattern table is not ended, investigate whether the value of the item "a primary control program" of the extracted machine configuration information and the value of an item "a primary control program" with the "pattern ID" under inspection have agreed (Step 1203).

[0071] (4) In Step 1202, when inspection of the machine configuration information extracted about all the "patterns ID" of an application execution condition pattern table is ended, end processing.

[0072] (5) In Step 1203, when the value of the item "a primary control program" of the extracted machine configuration information and the value of an item "a primary control program" with the "pattern ID" under inspection have agreed, investigate whether it is beyond the value of the item "the amount of memory requirements" in which the value of the item "the amount of empty memory" of the extracted machine configuration information has the "pattern ID" under inspection (Step 1204).

[0073] (6) In Step 1203, when the value of the item "a primary control program" of the extracted machine configuration information and the value of an item "a primary control program" with the "pattern ID" under inspection have not agreed, return to Step 1202.

[0074] (7) In Step 1204, when the value of the item "the amount of empty memory" of the extracted machine configuration information is beyond a value of an item "the amount of memory requirements" with the "pattern ID" under inspection, investigate whether it is beyond the value of the item "CPU" in which the value of the item "CPU" of the extracted machine configuration information has the "pattern ID" under inspection (Step 1205).

[0075] (8) In Step 1204, when the value of the item "the amount of empty memory" of the extracted machine configuration information is smaller than the value of an item "the amount of memory requirements" with the "pattern ID" under inspection, return to Step 1202.

[0076] (9) In Step 1205, when the value of the item "CPU" of the extracted machine configuration information is beyond a value of an item "CPU" with the "pattern ID" under inspection, create temporarily the extraction resulting table 1300 (drawing 13 of the after-mentioned for details), reproduce the information on each item with the "pattern ID" which was being made into the subject of examination from an application execution condition pattern table, and write it in the extraction resulting table 1300

(Step 1206).

[0077] (10) In Step 1205, when the value of the item "CPU" of the extracted machine configuration information is smaller than the value of an item "CPU" with the "pattern ID" under inspection, return to Step 1202.

[0078] (11) In Step 1205, when the value of the item "CPU" of the extracted machine configuration information is beyond a value of an item "CPU" with the "pattern ID" under inspection in an application execution condition pattern table, set initial value False as all the values of the item "the priority check n" (n=1-3) of the extraction resulting table 1300 after processing of Step 1206 (Step 1207).

[0079] (12) In Step 1205, when the value of the item "CPU" of the extracted machine configuration information is beyond a value of an item "CPU" with the "pattern ID" under inspection in an application execution condition pattern table, add 1 to the variable count showing the number of the program which can be performed after processing of Step 1207 (Step 1208), and return to Step 1202.

[0080] Drawing 13 is drawing which is prepared in order to write in the execution condition of only the program judged in the middle of the processing which extracts the program which was shown in aforementioned drawing 12, and which can be performed to be the program which can be performed and in which showing an example of the extraction resulting table 1300. The value of an item "a network address" located in the 1st step in aforementioned drawing 3 For example, "NCID00010", The value of "OS\_A 1.0" and an item "the amount of empty memory" "250KB", [ the value of an item "a primary control program" ] A user with the user-identification child "ritsuko" who is using the client terminal unit 110 in which the value of an item "CPU" has "180MHz" When loading of an application program A is required, using the information on the 1st step of the machine configuration information management data 300 extracted from the application execution condition pattern table 200 and the network address The value of the item "an application program A" which has 2 as 1 in the application execution condition pattern table 200 is extracted as an executable program, and is written in the extraction resulting table 1300 with the information on other items.

[0081] In drawing 13, the identification number (it is henceforth described as "Extraction ID") for 1301 managing an extraction resulting table for every train, and 201-205 show the operating condition of the application program reproduced from the application execution condition pattern table. 1302-1304 are items which show whether each program shown in the value of an item "an application program" fulfills the priority which the user of each element which determines the execution environment of application set up. "True" means fulfilling the priority which the user set up, and that "False" is not filling. As shown in Step 1207 in the first half, when the program which can be performed is extracted, all values are initialized by "False."

[0082] the number of attributes with which a user sets up the item of a priority check in the application execution-environment tuning data 400 as shown in 1302-1304 -- that is, three are prepared The value of an item "the priority check 1" shows whether the conditions which the user has set up are fulfilled about an attribute with the highest priority, and, as for the value of an item "the priority check 2", the value of an item "the priority check 3" shows, respectively whether the conditions which the user has set up about the low attribute of a priority most are fulfilled about an attribute with the 2nd highest priority. Therefore, the attributes of the priority which "priority check 1" - "the priority check 3" shows differ for every user who performs.

[0083] Drawing 14 is a flow chart explaining the procedure of "execution-time priority inspection processing" 1400 performed when two or more programs which can be performed exist in Step 1106 of aforementioned drawing 11. The execution-time priority inspection processing 1400 opens the extraction resulting table 1300 of (1) aforementioned drawing 13 (Step 1401).

[0084] (2) Extract the value of the application execution-environment tuning parameter which corresponds a user-identification child from the application execution-environment tuning data 400 at a key (Step 1402).

[0085] (3) In Step 1402, compare the value of each attribute of the extracted tuning parameter, and give a priority 1, a priority 2, and a priority 3 sequentially from an expensive attribute (Step 1403).

[0086] (4) Set initial value 1 as the variable n showing the height of the priority of the attribute under inspection (Step 1404).

[0087] (5) Investigate whether inspection was ended about all the "extraction ID" included in an extraction resulting table (Step 1405).

[0088] (6) In Step 1405, when inspection is ended about no the "extraction ID" included in an extraction resulting table, choose and perform a condition judging formula about the attribute of a priority n (Step 1500, Steps 1501-1505 of the after-mentioned [ a detail ]).

[0089] (7) In Step 1405, about all the "extraction ID" included in an extraction resulting table, when inspection is ended, end processing.

[0090] (8) In Step 1405, when inspection is not ended, as a result of processing Step 1500 about all the "extraction ID" included in an extraction resulting table, each item with the extraction ID under inspection investigates whether the condition judging formula was filled about the attribute of a priority n (Step 1407).

[0091] (9) In Step 1407, as a result of processing Step 1500, when each item with the "extraction ID" under inspection fills a condition judging formula about the attribute of a priority n, write "True" in the value of the item "the priority check n" of the extraction resulting table 1300 (Step 1408). The extraction resulting table after a processing end is later mentioned to drawing 17 .

[0092] (10) In Step 1407, as a result of processing Step 1500, when each item with the "extraction ID" under inspection does not fill a condition judging formula about the attribute of a priority n, return to Step 1405.

[0093] (11) In Step 1407, as a result of processing Step 1500, when each item with the "extraction ID" under inspection fills a condition judging formula about the attribute of a priority n, add 1 to the variable n showing the priority of the attribute made into processing of Step 1408, next a subject of examination, and, next, change an attribute to be examined into an attribute with a high priority (Step 1409).

[0094] (12) In Step 1407, as a result of processing Step 1500, when each item with the "extraction ID" under inspection fills a condition judging formula about the attribute of a priority n, investigate whether n is equal to 3 after processing of Step 1409, and investigate whether inspection was ended about all the attributes of a priority (Step 1410).

[0095] (13) In Step 1410, when inspection is ended about all attributes, return to Step 1405.

[0096] (14) In Step 1410, when inspection is ended about no attributes, return to Step 1500.

[0097] Drawing 15 is a flow chart which is performed about all the executable programs contained in an extraction resulting table when inspection is not ended and which explains the procedure of processing 1500 which chooses a condition judging formula and is performed about the attribute of a priority n in Step 1405 of aforementioned drawing 14 . The processing 1500 which chooses and performs a condition judging formula about the attribute of a priority n investigates whether the attribute of the priority n under (1) inspection is a "performance" (Step 1501).

[0098] (2) In Step 1501, when the attribute of the priority n under inspection is a "performance", choose and perform a "judgment formula\_performance" (drawing 16 of the after-mentioned for details) (Step 1502).

[0099] (3) In Step 1501, when the attribute of the priority n under inspection is not a "performance", investigate whether the attribute of the priority n under inspection is "memory" (Step 1503).

[0100] (4) In Step 1503, the attribute of the priority n under inspection is "memory."

[0101] A case chooses and performs "judgment formula\_memory" (drawing 16 of the after-mentioned for details) (Step 1504).

[0102] (5) In Step 1503, when the attribute of the priority n under inspection is not "memory", since the attribute of the priority n under inspection is a "version", it chooses and performs a "judgment formula\_version" (drawing 16 of the after-mentioned for details) (Step 1505).

[0103] Drawing 16 shows the concrete formula of the "judgment formula\_performance" shown in Steps 1502, 1504, and 1505 of aforementioned drawing 15 , "judgment formula\_memory", and a "judgment formula\_version." The program which was extracted in Step 1200 by the three aforementioned judgment formulas and which can be performed judges whether the application execution environment which a user wishes is realized. 1601 shows the formula of a "judgment formula\_performance." The numeric value to be used is four of "the values of "CPU" of the client terminal unit 110 which the user is using" extracted from "the value of the indispensable condition of the item "CPU" of an application execution condition pattern table", "the value of recommendation conditions", "the value of the item "a performance" of the application execution-environment tuning parameter which the user set up", and machine configuration information management data.

[0104] In aforementioned drawing 4 and drawing 5 , the minimum value of a tuning parameter described

that it is set as 0 (when it does not give priority most), and the highest value was set as 1 (when it gives priority most). When the value of the tuning parameter of a "performance" is 1 in a "judgment formula\_performance", The value of the item "CPU" of the machine configuration information of the client terminal unit 110 It is required that the value of the recommendation conditions of the item "CPU" of an application execution condition pattern table should be fulfilled. When the value of a tuning parameter is 0, fulfilling the value of the indispensable condition of the item "CPU" of an application execution condition pattern table presupposes that it is required. If the value of "CPU" of the extracted machine information is more than "value +(difference of value of indispensable condition and recommendation conditions) x (value set up by tuning data) of an indispensable condition", it will judge with fulfilling the value of the tuning parameter which the user set up (1601).

[0105] Also in the "judgment formula\_memory" shown in 1602 like a "judgment formula\_performance" When the value of the tuning parameter of "memory" is 1, the value of the item "the amount of empty memory" of the machine configuration information of the client terminal unit 110 It is required that the value of the recommendation conditions of the item "the amount of memory requirements" of an application execution condition pattern table should be fulfilled. When the value of a tuning parameter is 0, fulfilling the value of the indispensable condition of the item "the amount of memory requirements" of an application execution condition pattern table presupposes that it is required. If the value of the "amount of empty memory" of the extracted machine information is more than "value +(difference of value of indispensable condition and recommendation conditions) x (value set up by tuning data) of an indispensable condition", it will judge with fulfilling the value of the tuning parameter which the user set up (1602).

[0106] 1603 shows the formula of a "judgment formula\_version" chosen when the priority is most set as the newness of the version of the application program performed with the client terminal unit 110 highly. Since indispensable condition or recommendation conditions do not exist in a version unlike the two aforementioned judgment formulas, either, the data to be used are the "extraction ID" made into the value of the priority of the "version" set up by tuning data, the number of the program which can be performed, and the subject of examination.

[0107] The total number of trains of the extraction resulting table 1300 generated after execution of the processing 1200 which extracts all the programs that can perform aforementioned drawing 12 is used. When the value of the tuning parameter of a "version" is 1 Other attributes in an application execution condition pattern table "the amount of memory", As long as the value of the indispensable condition of "CPU" is fulfilled, and a program with the newest version, Namely, it is required that a program with the largest value of "Extraction ID" should be chosen, and when the value of a tuning parameter is 0 It is supposed that choosing most the old program of a version, i.e., a program with the smallest value 1 of "Extraction ID", is required. "The extraction ID made into the subject of examination" is more than  $10 \times (\text{value of "version" of extracted tuning data}) / (\text{number of the program which can be performed})$ . And when it is below  $10 \times (\text{value of "version" of extracted tuning data}) / (\text{number of program which can be performed}) + 1$ , it judges with fulfilling the value of the tuning parameter of a "version" for which the user wished (1603).

[0108] Drawing 17 is an example of the extraction resulting table 1300 after execution of the execution-time priority inspection processing 1400 shown in aforementioned drawing 14 . In the example given at the time of explanation of aforementioned drawing 13 , after creating an extraction resulting table Furthermore, the application execution-environment tuning data 400 of aforementioned drawing 4 are searched for a user-identification child "ritsuko" to a key. The value of the corresponding tuning parameter which exists in the 1st step in aforementioned drawing 4 is referred to. If processing which chooses the application program which realizes the operating environment which a user wishes from the program which can be performed is performed according to the judgment formula shown in aforementioned drawing 15 and drawing 16 , as the aforementioned example shows to 1701-1703 "True" is written in item "priority check 1" - "the priority check 3" whose the "extraction ID" of the extraction resulting table 1300 is "1." The aforementioned example shows that "Ver1.0" which is the value of the item "an application program A" which has "1" in "Extraction ID" is an execute permission, and is the optimal program which realizes the operating environment which a user wishes at the time of execution.

[0109] It can be said that "Ver1.1" which is the value of the item "an application program A" which has "2" in "Extraction ID" is not the optimal program which should be transmitted for a user with the hope to

which all the values of item "priority check 1" - "the priority check 3" continue being initial value "False", and give priority to a "performance" most among three attributes at the time of execution of an application program although it is an execute permission and a version is

[0110] However, as shown in drawing 17, a bird clapper does not surely have all the values from the item "the priority check 1" to "the priority check 3" in "True." Moreover, when two or more programs to which all attributes are set to "True" exist like "Ver1.0" of the program A which has "1" in "Extraction ID", it thinks. Even if it calculates by choosing a judgment formula with the execution-time priority which the user set up, it is not necessarily surely decided that the program which should be transmitted will be a meaning. Therefore, "transport-agent terminal-decision processing" briefly explained in Step 1800 of aforementioned drawing 11 is needed. A detail is explained below.

[0111] In Step 1104 of aforementioned drawing 11, when two or more programs which can be performed exist, drawing 18 is a flow chart explaining the procedure of "transport-agent terminal-decision processing" 1800 to perform, after performing execution-time priority inspection processing 1400 of aforementioned drawing 14. The transport-agent terminal-decision processing 1800 investigates whether inspection was ended about all the "extraction ID" of (1) extraction resulting table 1700 (Step 1801).

[0112] (2) In Step 1801, when inspection is ended about no the "extraction ID", investigate whether the value of an item "the priority check 1" with the "extraction ID" under inspection is "True" (Step 1802).

[0113] (3) In Step 1802, when the value of an item "the priority check 1" with the "extraction ID" under inspection is "True", turn ON the candidate flag 1 (Step 1803).

[0114] (4) In Step 1802, when the value of an item "the priority check 1" with the "extraction ID" under inspection is "False", return to Step 1801.

[0115] (5) In Step 1802, when the value of an item "the priority check 1" with the "extraction ID" under inspection is "True", investigate whether the value of an item "the priority check 2" with the extraction ID after processing of Step 1803 and under inspection is "True" (Step 1804).

[0116] (6) In Step 1804, when the value of an item "the priority check 2" with the "extraction ID" under inspection is "True", turn ON the candidate flag 2 (Step 1805).

[0117] (7) In Step 1804, when the value of an item "the priority check 2" with the "extraction ID" under inspection is "False", return to Step 1801.

[0118] (8) In Step 1804, when the value of an item "the priority check 2" with the "extraction ID" under inspection is "True", investigate whether the value of an item "the priority check 3" with the "extraction ID" after processing of Step 1805 and under inspection is "True" (Step 1806).

[0119] (9) In Step 1806, when the value of an item "the priority check 3" with the "extraction ID" under inspection is "True", determine the value of an item "an application program" with the "extraction ID" under inspection as a transport agent (Step 1807).

[0120] (10) In Step 1806, when the value of an item "the priority check 2" with the "extraction ID" under inspection is "False", return to Step 1801.

[0121] (11) In Step 1801, when inspection is ended about all the "extraction ID", the extraction ID all whose values of an item "the priority check n" are "True(s)" performs "selection exception handling" in order to show that it is not contained in an extraction resulting table (Step 1900, Steps 1901-1907 of the after-mentioned [ a detail ]). A detail is explained below.

[0122] Drawing 19 is a flow chart explaining the procedure of the selection exception handling 1900 performed when inspection is ended about all extraction ID of the extraction resulting table 1700 in Step 1801 of aforementioned drawing 18. The selection exception handling 1900 investigates whether the "extraction ID" to which (1) candidate flag 1 is turned on [ it ] exists (Step 1901).

[0123] (2) In Step 1901, when the "extraction ID" to which the candidate flag 1 is turned on [ it ] exists, investigate whether inspection was ended about all the "extraction ID" to which the candidate flag 1 is turned on [ it ] (Step 1902).

[0124] (3) In Step 1902, when inspection is ended about all the "extraction ID" to which the candidate flag 1 is turned on [ it ], determine the value of the item "an application program" which has the small "extraction ID" of a value most as a transport agent (Step 1903).

[0125] (4) In Step 1902, when inspection is ended about no the "extraction ID" to which the candidate flag 1 is turned on [ it ], investigate whether the candidate flag 2 of the "extraction ID" under inspection is ON (Step 1904).

[0126] (5) In Step 1904, when the candidate flag 2 of the "extraction ID" under inspection is turned on [ it



], determine the value of an item "an application program" with the "extraction ID" under inspection as a transport agent (Step 1905).

[0127] (6) In Step 1901, when the "extraction ID" to which the candidate flag 1 is turned on [ it ] does not exist, the value of the attribute "a version" of the application execution-environment tuning data 400 investigates whether it is smaller than the value of an attribute "a performance" (Step 1906).

[0128] (7) Determine the value of the item "an application program" in which it has the smallest "extraction ID" when the value of the attribute "a version" of the application execution-environment tuning data 400 is smaller than the value of an attribute "a performance" as a transport agent in Step 1906 (Step 1903).

[0129] (8) Determine the value of the item "an application program" in which it has the largest "extraction ID" when the value of the attribute "a version" of the application execution-environment tuning data 400 is larger than the value of an attribute "a performance" as a transport agent in Step 1906 (Step 1907).

[0130] Drawing 20 investigates the combination of the primary control program which can be performed with the client terminal unit 110 when the program which can be performed does not exist in Step 1105 of aforementioned drawing 11, and a program with the load demand, and only when the combination which can be performed exists and a user wishes, it is a flow chart explaining the procedure of the primary-control-program resending processing 2000 which transmits a primary control program and an application program. The primary-control-program resending processing 2000 investigates whether inspection was ended about all the "patterns ID" of (1) application execution condition pattern table 200 (Step 2001).

[0131] (2) In Step 2001, when inspection is ended about no the "patterns ID" of the application execution condition pattern table 200, investigate whether the value of the item "CPU" of the extracted machine configuration information is beyond a value of the indispensable condition of "CPU" of an application execution condition pattern table (Step 2002).

[0132] (3) In Step 2001, when inspection is ended about all the "patterns ID" of the application execution condition pattern table 200, it is the application which carried out program request into server equipment, and transmit the error message which shows that the program which can be performed does not exist to the client terminal unit 110 (Step 2008).

[0133] (4) In Step 2002, when the value of the item "CPU" of the extracted machine configuration information is beyond a value of the indispensable condition of "CPU" of an application execution condition pattern table, investigate whether the value of the item "the amount of empty memory" of the extracted machine configuration information is beyond a value of the indispensable condition of the "amount of memory requirements" of an application execution condition pattern table (Step 2003).

[0134] (5) In Step 2002, when the value of the item "CPU" of the extracted machine configuration information is smaller than the value of the indispensable condition of "CPU" of an application execution condition pattern table, it returns to Step 2001.

[0135] (6) In Step 2003, when the value of the item "the amount of empty memory" of the extracted machine configuration information is beyond a value of the indispensable condition of the "amount of memory requirements" of an application execution condition pattern table, transmit the message which stimulates resending of a primary control program to a client machine (Step 2004).

[0136] (7) In Step 2003, when the value of the item "the amount of empty memory" of the extracted machine configuration information is smaller than the value of the indispensable condition of the "amount of memory requirements" of an application execution condition pattern table, return to Step 2001.

[0137] (8) In Step 2003, when the value of the item "the amount of empty memory" of the extracted machine configuration information is beyond a value of the indispensable condition of the "amount of memory requirements" of an application execution condition pattern table, wait for a user's response after processing of Step 2004 (Step 2005).

[0138] (9) In Step 2003, when the value of the item "the amount of empty memory" of the extracted machine configuration information is beyond a value of the indispensable condition of the "amount of memory requirements" of an application execution condition pattern table, investigate whether the user wished resending of a primary control program after processing of Step 2005 (Step 2006).

[0139] (10) In Step 2006, when a user wishes resending of a primary control program, transmit the value



of an item "a primary control program" with the "extraction ID" under inspection in an application execution condition pattern table, and the value of an item "an application program" (Step 2007).

[0140] (11) In Step 2006, when a user does not wish resending of a primary control program, transmit the error message which shows that demanded application cannot be performed to the client terminal unit 110 with the present machine composition (Step 2008).

[0141] Drawing 21 is an example of the display screen of the message to which resending of the primary control program transmitted to the client terminal unit 110 is urged in Step 2004 of aforementioned drawing 20. If button 2101 "is and" is chosen, the primary control program which can perform the application program which carried out the load demand is transmitted with an application program, and a user can perform application which advanced the transfer request in the client terminal unit 110. If a button 2102 "no" is chosen, the error message which shows that the application program which carried out the load demand cannot be performed will be expressed as the present machine composition of the client terminal unit 110 which the user is using.

[0142]

[Effect of the Invention] It will be as follows if the effect acquired by the typical thing among invention indicated in this application is explained briefly.

[0143] (1) When the load demand of the application program is carried out from a client terminal unit From the hardware configuration information of the operating condition of application, and a client terminal unit With the execution-time priority of each element which determines the execution environment of application which extracted automatically what can be performed and was further set up by the user Since the program judged that is an execute permission in the execution environment optimal for a user is transmitted, a user Without being conscious of the hardware configuration information of the client terminal unit currently used, or the operating condition of application performing, the application program which can be performed can be loaded and it can perform in a comfortable operating environment.

[0144] (2) If it is the situation which cannot be performed on a client terminal unit, the application program which carried out the load demand It investigates whether there is any combination of the primary control program which can be automatically performed at a client terminal, and application. The screen which asks a user whether fix the execution environment of application by changing the machine composition of a client terminal unit in use is displayed. Only when a user wishes, in order to transmit a primary control program and an application program and to fix the execution environment of the demanded application program automatically on a client terminal unit, A user can get the execution environment of the application which required only at wishing the solution for performing on a client terminal unit while using the program which cannot carry out load and go, without devising by himself. Moreover, since a user's hope is reflected, when you do not wish change of environment, reloading of a primary control program can also be refused and a user can respond flexibly by the operating condition.

[0145] Even if it reloads a primary control program, when the program which can be performed does not exist in (3) and (2) Since the error message which shows that the program which can be performed on a client terminal unit in use in server equipment does not exist is displayed on a client terminal, without transmitting, After load demand issue, without waiting for the transfer end of a program, at the client terminal which the demanded program is using, a user can recognize that it cannot perform and can avoid wasting time on loading of the program which cannot be performed.

[0146] (4) Since it has a means by which a user can set up the priority of each element which determines the execution environment of application, to program server equipment, a user or a system administrator can set up the priority of each element which determines the execution environment of application so that the optimal operating environment according to the operating condition may be built.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-282686

(43) 公開日 平成11年(1999)10月15日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 9/445

13/00

15/16

識別記号

3 5 1

3 5 5

3 7 0

F I

G 0 6 F 9/06

13/00

15/16

4 2 0 J

3 5 1 H

3 5 5

3 7 0 N

審査請求 未請求 請求項の数 4 O L (全 19 頁)

(21) 出願番号

特願平10-83243

(22) 出願日

平成10年(1998)3月30日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 川端 律子

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 里山 元章

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 五十嵐 里香

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

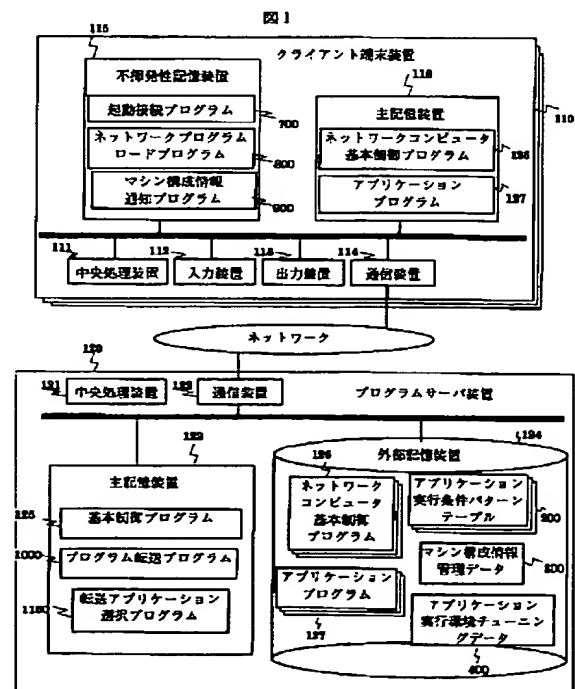
(74) 代理人 弁理士 小川 勝男

(54) 【発明の名称】 ネットワークコンピュータシステム

(57) 【要約】

【課題】 ユーザが、ハードウェア構成やアプリケーションの動作条件を意識することなく、ロード実行したいアプリケーションを、クライアント端末上で快適な動作環境にて実行する技術を提供する。

【解決手段】 クライアント端末よりプログラムのロード要求を受け取ったプログラムサーバ装置が、ロード要求のあったプログラムの動作条件と、クライアント端末のマシン構成情報と、プログラムの実行環境を決定する要素についてユーザ毎に設定された実行時優先度情報から、実行可能、かつユーザの希望する動作環境を実現する最適なプログラムを自動的に選択し、転送する。



**【特許請求の範囲】**

【請求項1】中央処理装置と主記憶装置と外部記憶装置とネットワーク通信装置とを備えたサーバ装置と、中央処理装置と主記憶装置とネットワーク通信装置とキーボードのごとき入力装置とディスプレイのごとき出力装置とを備えた、オフィス用コンピュータと、携帯型コンピュータと、情報検索サービス端末と、を含むクライアント端末装置との両方がネットワークで接続され、基本制御プログラムとアプリケーションプログラムを上記サーバ装置からネットワークを経由して、上記クライアント端末装置へロード実行するネットワークコンピュータシステムであって、

上記クライアント端末装置がロードするプログラムを格納するプログラムサーバ装置の外部記憶装置に、格納しているアプリケーションプログラムを実行するクライアント端末装置が満たすべき、「基本制御プログラム」、「必要メモリ量」、「性能」を定義したアプリケーション実行条件と、接続状態にある上記クライアント端末装置が持つ「基本制御プログラム」、「空きメモリ量」、「性能」を示すマシン構成情報と、プログラムの実行環境を決定する「性能」、「空きメモリ量」、「バージョン」の3つの要素についてユーザ毎に設定された実行時優先度情報と、を備え、上記アプリケーションプログラムの実行条件と、上記マシン構成情報と、ユーザ毎に設定された上記アプリケーション実行時の優先度情報から、クライアント端末装置にて実行可能、かつ実行時にユーザの希望する実行環境を実現する最適なプログラムを選択する手段を、上記プログラムサーバ装置に具備したことを特徴とするネットワークコンピュータシステム。

【請求項2】請求項1記載の実行可能、かつ最適なプログラムを選択する手段において、クライアント端末装置上で実行中の基本制御プログラムの変更を行えば、上記プログラムサーバ装置がクライアント端末装置に実行可能なプログラムを転送できる場合に、上記アプリケーションプログラムの実行条件と上記マシン構成情報から、実行可能な基本制御プログラムとアプリケーションプログラムの組み合わせを抽出し、使用中のクライアント端末装置の構成を変更することによりアプリケーションの実行環境を整えるか否かを、ユーザが使用時に指定できる手段を設け、ユーザが指定した場合のみに基本制御プログラムを再転送する処置により、アプリケーションプログラムを実行するための環境整備を自動的にこなうことを特徴とする請求項1記載のネットワークコンピュータシステム。

【請求項3】請求項1記載の実行可能、かつ最適なプログラムを自動的に選択する手段において、アプリケーションの実行環境をユーザ毎に決定する請求項1記載の3つの要素を持つ実行時優先度情報の中で、ユーザが最も優先度を高く設定した要素によって、異なる判定式を使

用し、ユーザの希望する実行環境を実現するプログラムの選択を行なうことを特徴とする請求項1記載のネットワークコンピュータシステム。

【請求項4】請求項1記載のアプリケーション実行時優先度情報において、アプリケーションプログラムの実行環境を決定する請求項1記載の3つの要素を持つ実行時優先度を、予めユーザ、あるいはシステム管理者が設定できる手段を有することを特徴とする請求項1記載のネットワークコンピュータシステム。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】本発明は、クライアント端末装置とサーバ装置との両方がネットワークで接続され、必要なプログラムを上記サーバ装置からネットワークを経由して、上記クライアント端末装置へロード実行するネットワークコンピュータシステムに適用して有効な技術に関するものである。

**【0002】**

【従来の技術】「日経エレクトロニクス1997. 4. 21 (no. 687)」に述べられているように、従来のネットワークコンピュータシステムでは、クライアント端末装置は起動時にサーバ装置と接続し、サーバ装置より基本制御プログラムをロード実行する。クライアント端末装置にてアプリケーションプログラムを実行する場合は、ユーザが実行したいアプリケーションプログラムの名前およびバージョンを指定して、サーバ装置に対しロード要求を発行する。サーバ装置は、クライアント端末装置から要求のあったアプリケーションプログラムを、外部記憶装置に格納されているアプリケーションプログラム群より検索し、クライアント端末装置に転送する機能を有する。

**【0003】**

【発明が解決しようとする課題】前記従来のネットワークコンピュータシステムでは、サーバ装置に格納されているアプリケーションプログラムに個々に動作条件を持つ様々なバージョンが混在し、実行可能であるか否かは、ロード要求を発行したクライアント端末装置の構成情報（ハードウェアや実行中の基本制御プログラム）との整合性に依存するため、実行可能なプログラムをロードするためには、ユーザが事前に、クライアント端末装置のマシン構成情報と、実行しようとするアプリケーションプログラムの動作条件との整合性を調べ、どのプログラム（バージョン）が実行可能であるかを判定する必要があった。

【0004】また、前述の整合性の判定の結果、使用中のクライアント端末装置の構成では実行可能なプログラムが存在しない場合、例えば基本制御プログラムのみが不整合である場合には、アプリケーションプログラムを実行するために、要求したアプリケーションと整合性のある基本制御プログラムをリロードすることにより環境

を整備するといった、要求したアプリケーションが実行不可能である要因の究明と解決策を、ユーザ自身が考案し、環境整備を手作業で行なわなければならなかった。

【0005】また、前述の整合性の判定に必要な情報を、ユーザが十分に把握していない場合、ロード要求を発行したプログラムが実行可能であるか否かは、一度クライアント端末装置にロード実行しない限り明確にならないため、ユーザは試行錯誤を繰り返し、実行不可能なプログラムのロードに時間を浪費することがあった。また、サーバ装置内に実行可能なプログラムが存在しない場合、すべてのプログラムをロード実行し、失敗に終わるまで、ユーザは、サーバ装置内に実行可能なプログラムが存在しないことを認識できなかった。

【0006】また、ロードしたアプリケーションプログラムが実行可能な場合においても、該アプリケーションプログラムのロード実行が、クライアント端末装置のメモリ不足や、性能の劣化を招くのでは、ユーザにとって快適な動作環境とは言えない。快適な動作環境とは、ユーザによって捉え方が異なり、性能の良さを示す場合も、複数のアプリケーションを同時に実行できる環境を示す場合も、アプリケーションプログラムの最新のバージョンを実行できる環境を示す場合もある。従来のネットワークコンピュータシステムでは、クライアント端末のマシン構成情報と、最初にロード実行したプログラムの実行条件から、プログラムの実行環境が必然的に決定され、ユーザが快適と感じる動作環境で実行できる保証がなかった。また、プログラムの実行環境をユーザが設定する手段も有していなかった。

【0007】本発明の目的は、前述した従来技術の問題点を解決し、クライアント端末装置からロード要求を受け取ったサーバ装置が、プログラムの転送前に、実行可能なプログラムを選択し、サーバ装置内にユーザの要求したアプリケーションであって、実行可能なプログラムが存在する場合においては、ユーザがマシン構成情報やプログラムの動作条件を意識することなく、要求したプログラムを確実に一度でロード実行することのできる、ネットワークコンピュータシステムを提供することにある。

【0008】本発明の目的は、ロード要求のあったクライアント端末装置上において基本制御プログラムの変更を行えば、サーバ装置が実行可能なアプリケーションプログラムを転送できる場合に、クライアント端末装置の構成を変更することによりアプリケーションの実行環境を整えるか否かを、ユーザが使用時に指定できる手段を設け、ユーザが指定した場合のみに要求したプログラムと整合性を持つ基本制御プログラムを再転送する処置により、ユーザが希望したアプリケーションの実行環境を得ることのできる、ネットワークコンピュータシステムを提供することにある。

【0009】本発明の目的は、基本制御プログラムを再

転送してもクライアント端末装置上にて実行可能なプログラムが存在しない場合に、要求したプログラムの転送を行わずに、エラーメッセージを表示し、サーバ装置内にユーザの要求したアプリケーションであって、使用中のクライアント端末装置上にて実行可能なプログラムは存在しないことをユーザに通知する、ネットワークコンピュータシステムを提供することにある。

【0010】本発明の目的は、ユーザのロード要求したアプリケーションであって、実行可能なプログラムが複数個存在する場合、実行可能なだけでなく、実行時にユーザの希望する動作環境を実現する最適なプログラムを選択し、ユーザに快適な動作環境を提供するネットワークコンピュータシステムを提供することにある。また、アプリケーションプログラムの実行環境を決定する「性能」、「空きメモリ量」、「バージョン」の3つの要素について、ユーザ、あるいはシステム管理者が、使用状況に応じた実行時優先度を、柔軟に設定できる手段を備え、ユーザが実行環境のチューニングを行なうことのできる、ネットワークコンピュータシステムを提供することにある。

【0011】

【課題を解決するための手段】本発明によれば前記目的は、クライアント端末装置に、起動後、プログラムサーバ装置と接続し、基本制御プログラムをロード実行する起動時接続手段と、プログラム転送要求をプログラムサーバ装置に対し発行し、プログラムを受信するネットワークプログラムロード手段と、クライアント端末装置にてハードウェア構成の変更が生じた時に、接続状態にあるプログラムサーバ装置に対し値の更新を通知するマシン構成情報通知手段と、を備え、プログラムサーバ装置には、ハードウェアの基本的な制御を行なうオペレーティングシステムのごときサーバ基本制御手段と、クライアント端末装置からのプログラムロード要求に応じてプログラムを転送するプログラム転送手段と、クライアント端末装置の要求に応じて転送するネットワークコンピュータ基本制御プログラム、およびアプリケーションプログラムと、該アプリケーションプログラムの動作条件を記したアプリケーション実行条件パターンテーブルと、接続状態にあるクライアント端末装置のハードウェア構成情報を管理するマシン構成情報管理データと、該アプリケーション実行条件パターンテーブルと、該マシン構成情報より、ユーザの要求したアプリケーションであって、サーバ装置内に存在する複数のプログラムの中から、実行可能なプログラムを抽出し、クライアント端末装置へ転送する、転送アプリケーション選択手段と、を備えることにより達成される。

【0012】該アプリケーション実行条件パターンテーブルは、該アプリケーションを実行するクライアント端末装置に要求する項目として、「基本制御プログラム」、「必要メモリ量」、「性能」の3つを有してお

り、「必要メモリ量」及び「性能」の項目の値についてのみに、必須条件（実行するために必ず満たさなければならない条件）と推奨条件（実行可能ではあるが、満たすことが望ましい条件）が設定される。該マシン構成情報管理データは、使用中のクライアント端末装置について、「基本制御プログラム」、「空きメモリ量」、「性能」の3つの項目を有している。

【0013】また、前記目的は、前記転送アプリケーション選択手段の処理途中において、クライアント端末装置上にて実行中の基本制御プログラムの変更を行なえば、サーバ装置が実行可能なアプリケーションプログラムを転送できる場合に、ロード要求を発行したクライアント端末装置のハードウェア構成情報を調べ、実行可能な基本制御プログラムと要求のあったアプリケーションプログラムの組み合わせを検索し、クライアント端末装置に要求したアプリケーションと整合性を持つ基本制御プログラムのリロードを促すメッセージを表示することにより、クライアントの構成情報を変更しプログラムの実行環境を整えるか否かを、使用時にユーザが指定できる手段を設け、ユーザが希望した場合のみに基本制御プログラムと、要求のあったアプリケーションプログラムとを、クライアント端末装置に転送し、ユーザが指定しない場合には、プログラムの転送を行わずに、エラーメッセージを表示し、使用中のマシン構成ではロード要求のあったプログラムが実行不可能であることをユーザに示すステップを、前記転送アプリケーション選択手段が含むことにより、達成される。

【0014】また、前記目的は、前記転送アプリケーション選択手段の処理途中において、実行しようとする該アプリケーション実行条件パターンテーブルの項目「性能」かつ項目「必要メモリ量」の必須条件の値を、該マシン構成情報管理データより抽出した、ユーザが使用中のクライアント端末装置のマシン構成情報の項目「性能」かつ項目「空きメモリ量」の値が満たしていない場合に、クライアント端末装置に、サーバ装置内に実行可能なプログラムが存在しないことを示す、エラーメッセージを表示するステップを、前記転送アプリケーション選択手段が含むことにより、達成される。

【0015】また、前記目的は、プログラムサーバ装置に、アプリケーションの実行環境を決定する「性能」、「空きメモリ量」、「バージョン」の各要素の実行時優先度をユーザ毎、あるいはセッション毎に示すアプリケーション実行環境チューニングデータを備え、該アプリケーション実行条件パターンテーブルと、該マシン構成管理情報より抽出した実行可能なプログラムに対し、該アプリケーション実行環境チューニングデータを参照し、ユーザが設定した優先度の最も高い要素に適した判定式を選択、実行し、ユーザに快適な実行環境を提供する最適なアプリケーションプログラムの選択を行なうステップを、前記転送アプリケーション選択手段が備える

ことにより達成される。また、ユーザ、あるいはシステム管理者が、アプリケーションの実行環境を決定する前記3つの要素の実行時優先度を設定する手段を、プログラムサーバ装置に備えることにより達成される。

【0016】

【発明の実施の形態】以下に、本発明のネットワークコンピュータ管理方法及び装置の一実施形態を図面により詳細に説明する。

【0017】図1は、本実施形態のネットワークコンピュータ管理方法を実施する実施装置の概略構成を示す図である。図1において、110はクライアント端末装置、120はプログラムサーバ装置である。クライアント端末装置110と、プログラムサーバ装置120とはネットワークで接続されている。クライアント端末装置110は、中央処理装置111と、キーボードのごとき入力装置112と、ディスプレイのごとき出力装置113と、通信装置114と、不揮発性記憶装置115と、主記憶装置116とを有している。不揮発性記憶装置115には、起動接続プログラム700と、ネットワークプログラムロードプログラム800と、マシン構成情報通知プログラム900と、を有している。

【0018】プログラムサーバ装置120は、中央処理装置121と、通信装置122と、主記憶装置123と、外部記憶装置124とを有している。主記憶装置123には、基本制御プログラム125と、プログラム転送プログラム1000と、転送アプリケーション選択プログラム1100とが、外部記憶装置124より読み込まれ、中央処理装置121にて実行される。外部記憶装置124は、ネットワークコンピュータ基本制御プログラム126と、アプリケーションプログラム127と、アプリケーション実行条件パターンテーブル200と、マシン構成情報管理データ300と、アプリケーション実行環境チューニングデータ400と、を格納している。

【0019】プログラムサーバ装置120の有する外部記憶装置124に格納されているネットワークコンピュータ基本制御プログラム126と、アプリケーションプログラム127とは、本実施形態のネットワークコンピュータ管理方法によってクライアント端末装置110の有する主記憶装置116に読み込まれ、中央処理装置111によって実行される。

【0020】図2は、本実施形態のネットワークコンピュータ管理方法を実施する実施装置の外部記憶装置124に格納されている、アプリケーション実行条件パターンテーブル200の一例を示す図である。アプリケーション実行条件パターンテーブル200は、同じ外部記憶装置124に格納されているアプリケーションプログラム127を、クライアント端末装置110において実行する際に必要とされる動作環境を定義しており、アプリケーションプログラム毎に一つ用意されている。

【0021】図2は、アプリケーションプログラムAをクライアント端末装置110において実行する場合に、クライアント端末装置110が満たさなければならない動作環境を定義している。図2において、201はテーブルのデータを列毎に管理するためのパターン識別番号（以下「パターンID」と記す）である。202は、アプリケーションプログラムAの実行可能なプログラムの名前、すなわち存在するアプリケーションプログラムAのバージョンの名前を示す。「パターンID」の1番から順に、バージョンの古い順に並べられており、番号が大きくなるに従い、バージョンが新しいことを示す。

【0022】203～205は、アプリケーションプログラムAを実行しようとする時にクライアント端末装置110のハードウェア構成が満たしていなければならない条件を示す。203は、アプリケーションプログラムAを実行する時に、クライアント端末装置110上で実行されていることが要求される基本制御プログラムの名前とバージョンを示す。204は、アプリケーションプログラムAの実行時にクライアント端末装置110のハードウェア構成において、必要になるメモリ量を、205は、クライアント端末装置110が満たすべき中央処理装置111の性能、すなわちCPUの値を示す。

【0023】204、205に示したメモリ量とCPUの値に関しては、必須条件と推奨条件が存在する。必須条件は、該当するアプリケーションプログラムのバージョンを実行するために、クライアント端末装置110が必ず満たさなければならない条件である。推奨条件は、満たしていることが望ましい条件であり、他のアプリケーションと平行して実行する場合や性能面において、より快適にプログラムを実行するための条件である。

【0024】図3は、本実施形態のネットワークコンピュータ管理方法を実施する実施装置のマシン構成情報管理データ300の一例を示す図である。マシン構成情報管理データ300は、プログラムサーバ装置120と接続したクライアント端末装置110のハードウェア構成情報を格納している。図3において、301はプログラムサーバ装置120と接続したクライアント端末装置110の識別番号、302～304は、301によって識別されるクライアント端末装置のハードウェア構成情報を示す。302は、実行中の基本制御プログラムの名前及びバージョンを示す。303は空きメモリ量を示す。304は、中央処理装置111の性能、すなわち搭載しているCPUの値を示す。

【0025】図4は、本実施形態のネットワークコンピュータ管理方法を実施する実施装置のアプリケーション実行環境チューニングデータ400の一例を示す図である。アプリケーション実行環境チューニングデータ400は、クライアント端末装置110におけるアプリケーションプログラムの実行環境を決定する「性能」、「空きメモリ量」、「バージョン」の3つの属性の実行時優

先度情報をユーザ毎、あるいはセッション毎に格納している。

【0026】図4において、401は、プログラムサーバ装置120と接続したクライアント端末装置110を使用しているユーザ識別番号、あるいはセッション番号を示す。402は、アプリケーションプログラムの実行環境を決定するチューニングパラメータの値を示し、403～405に示す3つの属性を有する。

【0027】403は、アプリケーションプログラムを実行するクライアント端末装置110の性能の優先度、404は、アプリケーションプログラムを実行するクライアント端末装置110の空きメモリ量の優先度、405は、クライアント端末装置110において実行するアプリケーションプログラムのバージョンの新しさの優先度である。402において、3つの値の和は必ず1になるように設定され、値が大きい属性ほど、実行時の優先度が高いことを意味する。

【0028】図5は、図4に示したアプリケーション実行環境チューニングデータ400の設定画面の一例を示す図である。アプリケーション実行環境チューニングデータ400の設定対象者は、クライアント端末装置110を使用する個々のユーザ、あるいはセッションを管理しているシステム管理者である。

【0029】403～405の各属性について、501に示す設定バーが用意され、ユーザあるいはシステム管理者は、502に示すポイントを動かすことにより、各属性の優先度を設定する。ユーザの設定したポイントの位置は、処理において、最も優先する場合を1、最も優先しない場合を0とし、どこに置かれているかによって、0～1の値に計算され、前記図4のアプリケーション実行環境チューニングデータ400に記入される。2つ以上の属性において、ポイントを同じ位置に設定すること、すなわち優先度を同じ値に設定することは不可能であり、また3つの属性の値の和は必ず1になるように設定される。

【0030】本プログラムは、プログラムサーバ装置内のアプリケーションプログラムの一つとして格納されており、ネットワークコンピュータの起動時に、基本制御プログラムと同時にクライアント端末装置に転送される通常デスクトッププログラムと呼ばれている基本アプリケーションプログラム群に含まれており、ユーザは、クライアント端末装置の使用中に設定、および変更を行なうことができる。

【0031】図6は、クライアント端末装置110がプログラムサーバ装置120に発行する要求のデータ構造を示す図である。601は、プログラムサーバ装置120に接続したクライアント端末装置110の識別番号、602はクライアント端末装置110がプログラムサーバ装置へ発行する命令を示す。603は補足情報を示し、602に示される命令の種類により含む情報が異な

る。クライアント端末装置110がプログラムサーバ装置120に対し発行する要求には、以下(A)～(C)に示す3種類、「初期プログラム転送」(602-a)、「アプリケーション転送」(602-b)、「マシン構成情報更新」(602-c)が存在する。

【0032】(A)は、命令が「初期プログラム転送」の場合にクライアント端末装置110がプログラムサーバ装置120に発行する要求のデータ構造を示す。クライアント端末装置110は電源が入ったと同時に、

(A)のデータ構造を持った要求を組み立てる。補足情報603には、ユーザ情報とクライアント端末装置110のハードウェア構成情報がセットされる。

【0033】603-aはユーザ識別子、603-bはクライアント端末装置110の中央処理装置111の性能、603-cはクライアント端末装置110の空きメモリ量、603-dはクライアント端末装置110に最初にロードされるネットワークコンピュータ基本制御プログラム126の名前とバージョンを示す。起動時に送られる補足情報603が、前記図3のマシン構成情報管理テーブル300に書き込まれる。

【0034】(B)は、命令が「アプリケーション転送」の場合にクライアント端末装置110がプログラムサーバ装置120に発行する要求のデータ構造を示す。電源が入り、基本制御プログラム126がロードされた状態のクライアント端末装置110が、プログラムサーバ装置120の外部記憶装置124に格納されているアプリケーションプログラム127を実行したい場合に、(B)のデータ構造を持った要求を組み立てる。補足情報603-eは、ロードしたいアプリケーションプログラムの名前を示す。

【0035】(C)は、命令が「マシン構成情報更新」の場合にクライアント端末装置110がプログラムサーバ装置120に発行する要求のデータ構造を示す。電源が入り、基本制御プログラム、アプリケーションプログラムがロードされ、実行されている状態のクライアント端末装置110において、メモリの増設、あるいは基本制御プログラムのリロードにより、ハードウェア構成情報に変更が生じた場合に、クライアント端末装置110が発行する要求である。補足情報603-fは、変更のあったハードウェア構成情報の属性および変更後の値を示す。本要求の発行により、前記図3に示したマシン構成情報管理データ300における該当する情報が更新される。

【0036】図7は、クライアント端末装置110の不揮発性記憶装置115に格納されている、起動接続プログラム700の処理手順を説明するフローチャートである。起動接続プログラム700は、

(1)あらかじめ格納されているネットワークアドレス、ユーザ識別子、中央処理装置111の性能(CPUの値)、空きメモリ量、最初にロードする基本制御プロ

グラムの名前及びバージョンを含む、クライアント端末装置110のマシン構成情報を取得する(ステップ701)。

【0037】(2)ネットワークプログラムロードプログラム800を呼び出し、ステップ701にて取得した情報から「初期プログラム転送要求」を組み立て、プログラムサーバ装置120に対し要求を発行し、ネットワークコンピュータ基本制御プログラム127をロードする(ステップ800、詳細は後述のステップ801～805)。

【0038】(3)ロードしたプログラムを実行する(ステップ702)。

【0039】図8は、クライアント端末装置110の不揮発性記憶装置115に格納されている、ネットワークプログラムロードプログラム800の処理手順を説明するフローチャートである。ネットワークプログラムロードプログラム800は、

(1)前記図6に示したプログラム転送要求を組み立てる(ステップ801)。

【0040】(2)プログラムサーバ装置120に対し、ステップ801において組み立てたプログラム転送要求を発行する(ステップ802)。

【0041】(3)通信装置114を使ってプログラムサーバ装置120と接続する(ステップ803)。

【0042】(4)プログラム転送要求に含まれるプログラムを、プログラムサーバ装置120の有する外部記憶装置124から、主記憶装置116へ読み込む(ステップ804)。

【0043】(5)読み込んだプログラムデータを主記憶装置へ書き込む(ステップ805)。

【0044】図9は、クライアント端末装置110の不揮発性記憶装置115に格納されている、マシン構成情報通知プログラム900の処理手順を説明するフローチャートである。マシン構成情報通知プログラム900は、

(1)クライアント端末装置110のマシン構成情報を取得する(ステップ901)。マシン構成情報は、中央処理装置の性能を示す「CPU」、空きメモリ量を示す「空きメモリ量」、最初にロードされた基本制御プログラムの名前及びバージョンを示す「基本制御プログラム」の3種類がある。

【0045】(2)クライアント端末装置110の「CPU」に変更が生じたか調べる(ステップ902)。

【0046】(3)クライアント端末装置110の「空きメモリ量」に変更が生じたか調べる(ステップ903)。

【0047】(4)クライアント端末装置110の「基本制御プログラム」に変更が生じたか調べる(ステップ904)。

【0048】(5)ステップ902～904において変



更が生じた場合は、変更のあった属性と変更後の値より、「マシン構成情報更新要求」を組み立てる（ステップ905）。

【0049】（6）ステップ902～904において変更が生じた場合は、ステップ905において組み立てた「マシン構成情報更新要求」を、プログラムサーバ装置120に対し発行する（ステップ906）。

【0050】（7）ステップ902～904において変更が生じなかった場合は、処理を終了する。

【0051】図10は、プログラムサーバ装置120の主記憶装置123に格納されているプログラム転送プログラム1000の処理手順を説明するフローチャートである。プログラム転送プログラム1000は、クライアント端末装置110の有するネットワークロードプログラム800からネットワークを介して転送される要求を処理する。プログラム転送プログラム1000は、

（1）クライアント端末装置110から発行された要求を、通信装置122を使用して受信する（ステップ1001）。

【0052】（2）ステップ1001において受信した要求の命令を調べる（ステップ1002）。

【0053】（3）ステップ1001において受信した要求の命令が「マシン構成情報更新」の場合、要求に含まれるネットワークアドレスよりマシン構成情報管理データ300において該当するマシン構成情報を抽出し、要求に含まれる補足情報を読み込み、変更のあった属性の値を更新する（ステップ1003）。

【0054】（4）ステップ1001において受信した要求の命令が「プログラム転送」の場合、要求に含まれるネットワークアドレスとアプリケーションプログラム名、およびユーザ識別子より、マシン構成情報管理データ300とアプリケーション実行条件パターンテーブル200とアプリケーション実行環境チューニングデータ400より必要な情報を抽出し、実行可能、かつユーザに快適な動作環境を提供する、転送すべき最適なアプリケーションプログラムを選択する（ステップ1100、詳細は後述のステップ1101～1107）。

【0055】（5）ステップ1001において受信した要求の命令が「プログラム転送」の場合、ステップ1100において選択したアプリケーションプログラムを転送する（ステップ1005）。

【0056】（6）ステップ1001において受信した要求の命令が「初期プログラム転送」の場合、要求に含まれるネットワークアドレス601と補足情報603を読み込み、マシン構成情報を作成する（ステップ1006）。

【0057】（7）ステップ1001において受信した要求の命令が「初期プログラム転送」の場合、1006において作成したマシン構成情報を、マシン構成情報管理データ300に追加する（ステップ1007）。

【0058】（8）ステップ1001において受信した要求の命令が「初期プログラム転送」の場合、要求の補足情報603に含まれる、ロード要求のあったネットワークコンピュータ基本制御プログラム126を外部記憶装置124から検索し、クライアント端末装置110に転送する（ステップ1008）。

【0059】図11は、プログラムサーバ装置120の主記憶装置123に格納されている、転送アプリケーション選択プログラム1100の処理手順を説明するフローチャートである。転送アプリケーション選択プログラム1100は、

（1）マシン構成情報管理データ300を開く（ステップ1101）。

【0060】（2）クライアント端末装置110から発行された要求に含まれるネットワークアドレス601をキーに、マシン構成情報管理データ300を検索し、該当するマシン構成情報を抽出する（ステップ1102）。

【0061】（3）クライアント端末装置110から発行された要求に含まれるアプリケーションプログラムの名前603-dをキーに、該当するアプリケーション実行条件パターンテーブル200を検索し、開く（ステップ1103）。

【0062】（4）ステップ1102において抽出したクライアント端末装置110のマシン構成情報と、ステップ1103において開いたアプリケーション実行条件パターンテーブルに含まれる動作条件を照合し、転送要求を発行したクライアント端末装置110上において実行可能なアプリケーションプログラムを抽出する（ステップ1200、詳細は後述のステップ1201～1208）。

【0063】（5）ステップ1200の処理を行なった結果、実行可能なアプリケーションプログラムが存在するか調べ（ステップ1104）、存在した場合、さらに複数個存在するか調べる（ステップ1105）。

【0064】（6）ステップ1104において、ステップ1200の処理を行なった結果、実行可能なアプリケーションプログラムが存在しなかった場合、クライアント端末装置110上で実行中のネットワークコンピュータ基本制御プログラムさえ入れ替えれば、動作可能であるアプリケーションプログラムが存在するかを調べ、存在した場合にはユーザが希望した場合にのみ基本制御プログラムを再送する“基本制御プログラム再送処理”を行なう（ステップ2000、詳細は後述のステップ2001～2008）。

【0065】（7）ステップ1105において、実行可能なアプリケーションプログラムが複数個存在しなかった場合、ステップ1200において抽出した実行可能なプログラムを転送プログラムとして決定する（ステップ1106）。

【0066】(8)ステップ1105において、実行可能なアプリケーションプログラムが複数個存在した場合、ユーザ識別子、あるいはセッション番号をキーにアプリケーション実行環境チューニングデータ400を検索し、該当するチューニングデータを参照し、ロード要求を行なったユーザに快適な動作環境を提供するプログラムを、ステップ1200において抽出済の実行可能なプログラムより選択する、実行時優先度検査処理1400を行なう(ステップ1400、詳細は後述の1401～1410)。

【0067】(9)ステップ1105において、実行可能なアプリケーションプログラムが複数個存在した場合、実行時優先度検査処理1400を実行した後、さらにユーザの希望する動作環境を実現するプログラムが複数個存在した場合、あるいはすべての属性においてユーザの希望を満たすプログラムが存在しなかった場合、において転送すべきプログラムを決定する、転送プログラム最終決定処理1800を行なう(ステップ1800、詳細は後述のステップ1801～1808)。

【0068】図12は、前記図11のステップ1103を行なった後に、実行可能なプログラムの抽出を行なう処理1200の処理手順を説明するフローチャートである。実行可能なプログラムを抽出する処理1200は、(1)実行可能なプログラムの個数を表す変数countに初期値0を設定する(ステップ1201)。

【0069】(2)アプリケーション実行条件パターンテーブルのすべての「パターンID」について、抽出したマシン構成情報の検査を終了したか調べる(ステップ1202)。

【0070】(3)ステップ1202において、アプリケーション実行条件パターンテーブルのすべての「パターンID」について、抽出したマシン構成情報の検査を終了していない場合、抽出したマシン構成情報の項目「基本制御プログラム」の値と、検査中の「パターンID」を持つ項目「基本制御プログラム」の値が合致しているかを調べる(ステップ1203)。

【0071】(4)ステップ1202において、アプリケーション実行条件パターンテーブルのすべての「パターンID」について、抽出したマシン構成情報の検査を終了した場合、処理を終了する。

【0072】(5)ステップ1203において、抽出したマシン構成情報の項目「基本制御プログラム」の値と、検査中の「パターンID」を持つ項目「基本制御プログラム」の値が合致している場合、抽出したマシン構成情報の項目「空きメモリ量」の値が、検査中の「パターンID」を持つ項目「必要メモリ量」の値以上であるかを調べる(ステップ1204)。

【0073】(6)ステップ1203において、抽出したマシン構成情報の項目「基本制御プログラム」の値と、検査中の「パターンID」を持つ項目「基本制御プ

ログラム」の値が合致していなかった場合、ステップ1202に戻る。

【0074】(7)ステップ1204において、抽出したマシン構成情報の項目「空きメモリ量」の値が、検査中の「パターンID」を持つ項目「必要メモリ量」の値以上であった場合、抽出したマシン構成情報の項目「CPU」の値が、検査中の「パターンID」を持つ項目「CPU」の値以上であるかを調べる(ステップ1205)。

【0075】(8)ステップ1204において、抽出したマシン構成情報の項目「空きメモリ量」の値が、検査中の「パターンID」を持つ項目「必要メモリ量」の値より小さかった場合、ステップ1202に戻る。

【0076】(9)ステップ1205において、抽出したマシン構成情報の項目「CPU」の値が、検査中の「パターンID」を持つ項目「CPU」の値以上であった場合、抽出結果テーブル1300(詳細は後述の図13)を一時的に作成し、検査対象としていた「パターンID」を持つ各項目の情報を、アプリケーション実行条件パターンテーブルより複製し、抽出結果テーブル1300に書き込む(ステップ1206)。

【0077】(10)ステップ1205において、抽出したマシン構成情報の項目「CPU」の値が、検査中の「パターンID」を持つ項目「CPU」の値より小さかった場合、ステップ1202に戻る。

【0078】(11)ステップ1205において、抽出したマシン構成情報の項目「CPU」の値が、アプリケーション実行条件パターンテーブル内の検査中の「パターンID」を持つ項目「CPU」の値以上であった場合、ステップ1206の処理後、抽出結果テーブル1300の項目「優先度チェックn」(n=1～3)のすべての値に初期値Falseを設定する(ステップ1207)。

【0079】(12)ステップ1205において、抽出したマシン構成情報の項目「CPU」の値が、アプリケーション実行条件パターンテーブル内の検査中の「パターンID」を持つ項目「CPU」の値以上であった場合、ステップ1207の処理後、実行可能なプログラムの個数を表す変数countに1を加え(ステップ1208)、ステップ1202に戻る。

【0080】図13は、前記図12に示した実行可能なプログラムを抽出する処理の途中において、実行可能なプログラムと判定されたプログラムのみの実行条件を書き込むために用意される、抽出結果テーブル1300の一例を示す図である。例えば、前記図3において第1段に位置する、項目「ネットワークアドレス」の値が“NCID00010”、項目「基本制御プログラム」の値が“OS\_A1.0”、項目「空きメモリ量」の値が“250KB”、項目「CPU」の値が“180MHz”を持つクライアント端末装置110を使用しているユーザ識別子“rits

uko”を持つユーザが、アプリケーションプログラムAのロードを要求した場合、アプリケーション実行条件パターンテーブル200とネットワークアドレスより抽出されたマシン構成情報管理データ300の第1段の情報により、アプリケーション実行条件パターンテーブル200において「パターンID」1と2を持つ項目「アプリケーションプログラムA」の値が、実行可能プログラムとして抽出され、他の項目の情報と共に抽出結果テーブル1300に書き込まれる。

【0081】図13において、1301は抽出結果テーブルを列毎に管理するための識別番号（以後、「抽出ID」と記す）、201～205は、アプリケーション実行条件パターンテーブルより複製したアプリケーションプログラムの動作条件を示す。1302～1304は、アプリケーションの実行環境を決定する各要素の、ユーザが設定した優先度を項目「アプリケーションプログラム」の値に示される各プログラムが、満たしているかを示す項目である。“True”はユーザの設定した優先度を満たしていること、“False”は満たしていないことを意味する。前期ステップ1207に示したように、実行可能なプログラムを抽出した時点では、すべての値が“False”で初期化される。

【0082】1302～1304に示すように、優先度チェックの項目は、アプリケーション実行環境チューニングデータ400においてユーザが設定する属性の数、すなわち3つ用意されている。項目「優先度チェック1」の値は、最も優先度の高い属性について、ユーザが設定している条件を満たしているかを示し、項目「優先度チェック2」の値は2番目に優先度の高い属性について、項目「優先度チェック3」の値は最も優先度の低い属性について、ユーザが設定している条件を満たしているかをそれぞれ示す。したがって、「優先度チェック1」～「優先度チェック3」の示す優先度の属性は、実行するユーザ毎に異なる。

【0083】図14は、前記図11のステップ1106において、実行可能なプログラムが複数個存在した場合に実行される、“実行時優先度検査処理”1400の処理手順を説明するフローチャートである。実行時優先度検査処理1400は、

(1) 前記図13の抽出結果テーブル1300を開く（ステップ1401）。

【0084】(2) ユーザ識別子をキーに、アプリケーション実行環境チューニングデータ400より、該当するアプリケーション実行環境チューニングパラメータの値を抽出する（ステップ1402）。

【0085】(3) ステップ1402において、抽出したチューニングパラメータの各属性の値を比較し、値の高い属性から順に優先度1、優先度2、優先度3をつける（ステップ1403）。

【0086】(4) 検査中の属性の優先度の高さを表す

変数nに、初期値1を設定する（ステップ1404）。

【0087】(5) 抽出結果テーブルに含まれるすべての「抽出ID」について、検査を終了したか調べる（ステップ1405）。

【0088】(6) ステップ1405において、抽出結果テーブルに含まれるすべての「抽出ID」について、検査を終了していない場合、優先度nの属性について、条件判定式を選択し、実行する（ステップ1500、詳細は後述のステップ1501～1505）。

【0089】(7) ステップ1405において、抽出結果テーブルに含まれるすべての「抽出ID」について、検査を終了した場合、処理を終了する。

【0090】(8) ステップ1405において、抽出結果テーブルに含まれるすべての「抽出ID」について、検査を終了していない場合、ステップ1500の処理を行なった結果、検査中の抽出IDを持つ各項目が、優先度nの属性について条件判定式を満たしたかを調べる（ステップ1407）。

【0091】(9) ステップ1407において、ステップ1500の処理を行なった結果、検査中の「抽出ID」を持つ各項目が、優先度nの属性について条件判定式を満たした場合、抽出結果テーブル1300の項目「優先度チェックn」の値に“True”を書き込む（ステップ1408）。処理終了後の抽出結果テーブルを図17に後述する。

【0092】(10) ステップ1407において、ステップ1500の処理を行なった結果、検査中の「抽出ID」を持つ各項目が、優先度nの属性について条件判定式を満たさなかった場合、ステップ1405に戻る。

【0093】(11) ステップ1407において、ステップ1500の処理を行なった結果、検査中の「抽出ID」を持つ各項目が、優先度nの属性について条件判定式を満たした場合、ステップ1408の処理の次に、検査対象とする属性の優先度を表す変数nに1を加え、検査対象の属性を、次に優先度の高い属性に変える（ステップ1409）。

【0094】(12) ステップ1407において、ステップ1500の処理を行なった結果、検査中の「抽出ID」を持つ各項目が、優先度nの属性について条件判定式を満たした場合、ステップ1409の処理の後に、nが3に等しいかを調べ、優先度のすべての属性について検査を終了したかを調べる（ステップ1410）。

【0095】(13) ステップ1410において、すべての属性について検査を終了していた場合には、ステップ1405に戻る。

【0096】(14) ステップ1410において、すべての属性について検査を終了していなかった場合には、ステップ1500に戻る。

【0097】図15は、前記図14のステップ1405において、抽出結果テーブルに含まれるすべての実行可

能プログラムについて、検査を終了していない場合に実行される、優先度nの属性について条件判定式を選択し、実行する処理1500の処理手順を説明するフローチャートである。優先度nの属性について条件判定式を選択し、実行する処理1500は、

(1) 検査中の優先度nの属性が、「性能」であるかを調べる(ステップ1501)。

【0098】(2) ステップ1501において、検査中の優先度nの属性が「性能」である場合は、「判定式\_性能」(詳細は後述の図16)を選択し、実行する(ステップ1502)。

【0099】(3) ステップ1501において、検査中の優先度nの属性が「性能」でない場合は、検査中の優先度nの属性が「メモリ」であるかを調べる(ステップ1503)。

【0100】(4) ステップ1503において、検査中の優先度nの属性が「メモリ」である。

【0101】場合は、「判定式\_メモリ」(詳細は後述の図16)を選択し、実行する(ステップ1504)。

【0102】(5) ステップ1503において、検査中の優先度nの属性が「メモリ」でない場合は、検査中の優先度nの属性は「バージョン」であるため、「判定式\_バージョン」(詳細は後述の図16)を選択し、実行する(ステップ1505)。

【0103】図16は、前記図15のステップ1502、1504、および1505に示した「判定式\_性能」、「判定式\_メモリ」、「判定式\_バージョン」の具体的な計算式を示している。前記3つの判定式により、ステップ1200において抽出された実行可能なプログラムが、ユーザの希望するアプリケーション実行環境を実現するか否かを判定する。1601は「判定式\_性能」の計算式を示す。使用する数値は、「アプリケーション実行条件パターンテーブルの項目「CPU」の必須条件の値」と「推奨条件の値」、および「ユーザの設定したアプリケーション実行環境チューニングパラメータの項目「性能」の値」、およびマシン構成情報管理データより抽出した、「ユーザが使用しているクライアント端末装置110の「CPU」の値」の4つである。

【0104】前記図4、図5において、チューニングパラメータの最低値(最も優先しない場合)は0に設定され、最高値(最も優先する場合)は1に設定されると記した。「判定式\_性能」において、「性能」のチューニングパラメータの値が1の場合、クライアント端末装置110のマシン構成情報の項目「CPU」の値が、アプリケーション実行条件パターンテーブルの項目「CPU」の推奨条件の値を満たすことが要求され、チューニングパラメータの値が0の場合、アプリケーション実行条件パターンテーブルの項目「CPU」の必須条件の値を満たすことが要求されるとし、抽出したマシン情報の「CPU」の値が、「必須条件の値+ (必須条件と推奨

条件の値の差) × (チューニングデータで設定された値)」以上であれば、ユーザの設定したチューニングパラメータの値を満たすと判定する(1601)。

【0105】1602に示す「判定式\_メモリ」においても「判定式\_性能」と同様に、「メモリ」のチューニングパラメータの値が1の場合、クライアント端末装置110のマシン構成情報の項目「空きメモリ量」の値が、アプリケーション実行条件パターンテーブルの項目「必要メモリ量」の推奨条件の値を満たすことが要求され、チューニングパラメータの値が0の場合、アプリケーション実行条件パターンテーブルの項目「必要メモリ量」の必須条件の値を満たすことが要求されるとし、抽出したマシン情報の「空きメモリ量」の値が、「必須条件の値+ (必須条件と推奨条件の値の差) × (チューニングデータで設定された値)」以上であれば、ユーザの設定したチューニングパラメータの値を満たすと判定する(1602)。

【0106】1603は、クライアント端末装置110にて実行するアプリケーションプログラムのバージョンの新しさに、最も優先度が高く設定されていた場合に選択される、「判定式\_バージョン」の計算式を示す。前記2つの判定式と異なり、バージョンには必須条件も推奨条件も存在しないため、使用するデータは、チューニングデータで設定された「バージョン」の優先度の値、実行可能なプログラムの個数、検査対象としている「抽出ID」である。

【0107】前記図12の、実行可能なプログラムをすべて抽出する処理1200の実行後に生成される抽出結果テーブル1300の総列数を利用し、「バージョン」のチューニングパラメータの値が1の場合は、アプリケーション実行条件パターンテーブルにおける他の属性「メモリ量」、および「CPU」の必須条件の値を満たしている限り、最もバージョンの新しいプログラム、すなわち「抽出ID」の最も大きい値を持つプログラムを選ぶことが要求され、チューニングパラメータの値が0の場合は、最もバージョンの古いプログラム、すなわち「抽出ID」の最も小さい値1を持つプログラムを選ぶことを要求されるとし、「検査対象としている抽出ID」が『 $10 \times (\text{抽出したチューニングデータの「バージョン」の値}) \div (\text{実行可能なプログラムの個数})$ 』以上であり、かつ『 $10 \times (\text{抽出したチューニングデータの「バージョン」の値}) \div (\text{実行可能なプログラムの個数}) + 1$ 』以下である場合に、ユーザの希望した「バージョン」のチューニングパラメータの値を満たすと判定する(1603)。

【0108】図17は、前記図14に示した、実行時優先度検査処理1400の実行後の抽出結果テーブル1300の一例である。前記図13の説明時に挙げた例で、抽出結果テーブルを作成した後に、さらにユーザ識別子“ritsuko”をキーに、前記図4のアプリケーション実

行環境チューニングデータ400を検索し、前記図4において第1段に存在する該当するチューニングパラメータの値を参照し、前記図15、図16に示した判定式に従って、実行可能なプログラムから、ユーザの希望する動作環境を実現するアプリケーションプログラムを選択する処理を行なうと、前記例では1701~1703に示すように、抽出結果テーブル1300の「抽出ID」が「1」である項目「優先度チェック1」~「優先度チェック3」に“True”が書き込まれる。前記例では、「抽出ID」に「1」を持つ項目「アプリケーションプログラムA」の値である「Ver1.0」が、実行可能であり、かつ実行時にユーザの希望する動作環境を実現する最適のプログラムであることを示す。

【0109】「抽出ID」に「2」を持つ項目「アプリケーションプログラムA」の値である「Ver1.1」は、実行可能でありバージョンも新しいが、項目「優先度チェック1」~「優先度チェック3」の値がすべて初期値“False”のままであり、アプリケーションプログラムの実行時に3つの属性のうち「性能」を最も優先する希望を持つユーザにとっては、転送すべき最適のプログラムではないと言える。

【0110】しかし、図17に示すように、項目「優先度チェック1」から「優先度チェック3」までのすべての値が“True”になることが、必ずあるとは限らない。また「抽出ID」に「1」を持つプログラムAの「Ver1.0」のように、すべての属性が“True”になるプログラムが複数個存在する場合も考えられる。ユーザの設定した実行時優先度によって判定式を選択し、計算を行なっても、転送すべきプログラムが必ず一意に決まるとは限らない。したがって、前記図11のステップ1800において簡単に説明した「転送プログラム最終決定処理」が必要となる。以下に詳細を説明する。

【0111】図18は、前記図11のステップ1104において、実行可能なプログラムが複数個存在した場合に、前記図14の実行時優先度検査処理1400を行なった後に、実行する「転送プログラム最終決定処理」1800の処理手順を説明するフローチャートである。転送プログラム最終決定処理1800は、

(1) 抽出結果テーブル1700のすべての「抽出ID」について、検査を終了したか調べる(ステップ1801)。

【0112】(2) ステップ1801において、すべての「抽出ID」について、検査を終了していない場合、検査中の「抽出ID」を持つ項目「優先度チェック1」の値が“True”であるか調べる(ステップ1802)。

【0113】(3) ステップ1802において、検査中の「抽出ID」を持つ項目「優先度チェック1」の値が“True”であった場合、候補フラグ1をONにする(ステップ1803)。

【0114】(4) ステップ1802において、検査中

の「抽出ID」を持つ項目「優先度チェック1」の値が“False”であった場合、ステップ1801に戻る。

【0115】(5) ステップ1802において、検査中の「抽出ID」を持つ項目「優先度チェック1」の値が“True”であった場合、ステップ1803の処理の後、検査中の抽出IDを持つ項目「優先度チェック2」の値が“True”であるか調べる(ステップ1804)。

【0116】(6) ステップ1804において、検査中の「抽出ID」を持つ項目「優先度チェック2」の値が“True”であった場合、候補フラグ2をONにする(ステップ1805)。

【0117】(7) ステップ1804において、検査中の「抽出ID」を持つ項目「優先度チェック2」の値が“False”であった場合、ステップ1801に戻る。

【0118】(8) ステップ1804において、検査中の「抽出ID」を持つ項目「優先度チェック2」の値が“True”であった場合、ステップ1805の処理の後、検査中の「抽出ID」を持つ項目「優先度チェック3」の値が“True”であるか調べる(ステップ1806)。

【0119】(9) ステップ1806において、検査中の「抽出ID」を持つ項目「優先度チェック3」の値が“True”であった場合、検査中の「抽出ID」を持つ項目「アプリケーションプログラム」の値を転送プログラムとして決定する(ステップ1807)。

【0120】(10) ステップ1806において、検査中の「抽出ID」を持つ項目「優先度チェック2」の値が“False”であった場合、ステップ1801に戻る。

【0121】(11) ステップ1801において、すべての「抽出ID」について、検査を終了した場合、項目「優先度チェックn」のすべての値が“True”である抽出IDは抽出結果テーブル中に含まれないことを示すため、「選択例外処理」を行なう(ステップ1900、詳細は後述のステップ1901~1907)。以下に詳細を説明する。

【0122】図19は、前記図18のステップ1801において、抽出結果テーブル1700のすべての抽出IDについて、検査を終了した場合に実行する選択例外処理1900の処理手順を説明するフローチャートである。選択例外処理1900は、

(1) 候補フラグ1がONになっている「抽出ID」が存在するか調べる(ステップ1901)。

【0123】(2) ステップ1901において、候補フラグ1がONになっている「抽出ID」が存在した場合、候補フラグ1がONになっているすべての「抽出ID」について検査を終了したかを調べる(ステップ1902)。

【0124】(3) ステップ1902において、候補フラグ1がONになっているすべての「抽出ID」について検査を終了した場合、最も値の小さい「抽出ID」を持つ項目「アプリケーションプログラム」の値を、転送

プログラムとして決定する(ステップ1903)。

【0125】(4)ステップ1902において、候補フラグ1がONになっているすべての「抽出ID」について検査を終了していない場合、検査中の「抽出ID」の候補フラグ2がONであるか調べる(ステップ1904)。

【0126】(5)ステップ1904において、検査中の「抽出ID」の候補フラグ2がONになっている場合、検査中の「抽出ID」を持つ項目「アプリケーションプログラム」の値を、転送プログラムとして決定する(ステップ1905)。

【0127】(6)ステップ1901において、候補フラグ1がONになっている「抽出ID」が存在しなかった場合、アプリケーション実行環境チューニングデータ400の属性「バージョン」の値が、属性「性能」の値より小さいかを調べる(ステップ1906)。

【0128】(7)ステップ1906において、アプリケーション実行環境チューニングデータ400の属性「バージョン」の値が、属性「性能」の値より小さい場合、最も小さい「抽出ID」を持つ項目「アプリケーションプログラム」の値を転送プログラムとして決定する(ステップ1903)。

【0129】(8)ステップ1906において、アプリケーション実行環境チューニングデータ400の属性「バージョン」の値が、属性「性能」の値より大きい場合、最も大きい「抽出ID」を持つ項目「アプリケーションプログラム」の値を転送プログラムとして決定する(ステップ1907)。

【0130】図20は、前記図11のステップ1105において、実行可能なプログラムが存在しなかった場合に、クライアント端末装置110にて実行できる基本制御プログラムとロード要求のあったプログラムの組み合わせを調べ、実行できる組み合わせが存在し、かつユーザが希望した場合にのみ基本制御プログラムとアプリケーションプログラムを転送する、基本制御プログラム再送処理2000の処理手順を説明するフローチャートである。基本制御プログラム再送処理2000は、

(1)アプリケーション実行条件パターンテーブル200のすべての「パターンID」について、検査を終了したか調べる(ステップ2001)。

【0131】(2)ステップ2001において、アプリケーション実行条件パターンテーブル200のすべての「パターンID」について、検査を終了していない場合、抽出したマシン構成情報の項目「CPU」の値がアプリケーション実行条件パターンテーブルの「CPU」の必須条件の値以上であるかを調べる(ステップ2002)。

【0132】(3)ステップ2001において、アプリケーション実行条件パターンテーブル200のすべての「パターンID」について、検査を終了した場合、クラ

イアント端末装置110に、サーバ装置内にプログラム要求したアプリケーションであって、実行可能なプログラムは存在しないことを示す、エラーメッセージを送信する(ステップ2008)。

【0133】(4)ステップ2002において、抽出したマシン構成情報の項目「CPU」の値がアプリケーション実行条件パターンテーブルの「CPU」の必須条件の値以上であった場合、抽出したマシン構成情報の項目「空きメモリ量」の値がアプリケーション実行条件パターンテーブルの「必要メモリ量」の必須条件の値以上であるかを調べる(ステップ2003)。

【0134】(5)ステップ2002において、抽出したマシン構成情報の項目「CPU」の値が、アプリケーション実行条件パターンテーブルの「CPU」の必須条件の値より小さかった場合、ステップ2001に戻る。

【0135】(6)ステップ2003において、抽出したマシン構成情報の項目「空きメモリ量」の値がアプリケーション実行条件パターンテーブルの「必要メモリ量」の必須条件の値以上であった場合、基本制御プログラムの再送を促すメッセージをクライアントマシンに送信する(ステップ2004)。

【0136】(7)ステップ2003において、抽出したマシン構成情報の項目「空きメモリ量」の値がアプリケーション実行条件パターンテーブルの「必要メモリ量」の必須条件の値より小さかった場合、ステップ2001に戻る。

【0137】(8)ステップ2003において、抽出したマシン構成情報の項目「空きメモリ量」の値がアプリケーション実行条件パターンテーブルの「必要メモリ量」の必須条件の値以上であった場合、ステップ2004の処理後、ユーザの応答を待つ(ステップ2005)。

【0138】(9)ステップ2003において、抽出したマシン構成情報の項目「空きメモリ量」の値がアプリケーション実行条件パターンテーブルの「必要メモリ量」の必須条件の値以上であった場合、ステップ2005の処理後、ユーザが基本制御プログラムの再送を希望したか調べる(ステップ2006)。

【0139】(10)ステップ2006において、ユーザが基本制御プログラムの再送を希望した場合、アプリケーション実行条件パターンテーブル中の検査中の「抽出ID」を持つ項目「基本制御プログラム」の値と、項目「アプリケーションプログラム」の値を転送する(ステップ2007)。

【0140】(11)ステップ2006において、ユーザが基本制御プログラムの再送を希望しなかった場合、クライアント端末装置110に、現在のマシン構成では要求したアプリケーションを実行不可能であることを示す、エラーメッセージを送信する(ステップ2008)。

【0141】図21は、前記図20のステップ2004において、クライアント端末装置110に送信される基本制御プログラムの再送を促すメッセージの表示画面の一例である。ボタン2101「はい」を選択すれば、ロード要求したアプリケーションプログラムが実行できる基本制御プログラムが、アプリケーションプログラムと共に転送され、ユーザは転送要求を出したアプリケーションをクライアント端末装置110において実行できる。ボタン2102「いいえ」を選択すれば、ユーザの使用しているクライアント端末装置110の現在のマシン構成では、ロード要求したアプリケーションプログラムを実行不可能であることを示すエラーメッセージが表示される。

#### 【0142】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記のとおりである。

【0143】(1) クライアント端末装置からアプリケーションプログラムをロード要求した場合に、アプリケーションの動作条件とクライアント端末装置のハードウェア構成情報より、実行可能なものを自動的に抽出し、さらにユーザによって設定された、アプリケーションの実行環境を決定する各要素の実行時優先度により、ユーザにとって最適な実行環境にて実行可能であると判定したプログラムを転送するので、ユーザは、使用しているクライアント端末装置のハードウェア構成情報や実行したいアプリケーションの動作条件を意識することなく、実行可能なアプリケーションプログラムをロードし、快適な動作環境にて実行することができる。

【0144】(2) ロード要求したアプリケーションプログラムを、クライアント端末装置上にて実行できない状況であれば、自動的にクライアント端末にて実行できる基本制御プログラムとアプリケーションの組み合わせがあるかを調べ、使用中のクライアント端末装置のマシン構成を変更することによりアプリケーションの実行環境を整備するか否かをユーザに問い合わせる画面を表示し、ユーザが希望した場合のみに基本制御プログラムとアプリケーションプログラムを転送し、要求したアプリケーションプログラムの実行環境をクライアント端末装置上に自動的に整備するため、ユーザはロード実行できないプログラムを使用中のクライアント端末装置上にて実行するための解決策を、自分自身で考案することなく、希望するのみで要求したアプリケーションの実行環境を得ることができる。また、ユーザの希望が反映されるため、環境の変更を希望しない場合には、基本制御プログラムのリロードを拒否することもでき、ユーザは使用状況により柔軟に対応できる。

【0145】(3) (2)において、基本制御プログラムをリロードしても実行可能なプログラムが存在しない場合は、転送を行わずに、サーバ装置内に使用中のク

ライアント端末装置上で実行可能なプログラムが存在しないことを示す、エラーメッセージがクライアント端末に表示されるため、ユーザはロード要求発行後、プログラムの転送終了を待つことなく、要求したプログラムが使用中のクライアント端末では実行不可能であることを認識でき、実行不可能なプログラムのロードに時間を浪費することを回避できる。

【0146】(4) アプリケーションの実行環境を決定する各要素の優先度をユーザが設定できる手段をプログラムサーバ装置に有するので、ユーザ、あるいはシステム管理者は、使用状況に応じた最適な動作環境を構築するように、アプリケーションの実行環境を決定する各要素の優先度を設定することができる。

#### 【図面の簡単な説明】

【図1】本実施形態のネットワークコンピュータ管理方法を実施する実施装置の概略構成を示す図である。

【図2】本実施形態のネットワークコンピュータ管理方法を実施する実施装置のアプリケーション実行条件パターンテーブル200の一例を示す図である。

【図3】本実施形態のネットワークコンピュータ管理方法を実施する実施装置のマシン構成情報管理データ300の一例を示す図である。

【図4】本実施形態のネットワークコンピュータ管理方法を実施する実施装置のアプリケーション実行環境チューニングデータ400の一例を示す図である。

【図5】図4におけるアプリケーション実行環境チューニングデータの設定画面の一例を示す図である。

【図6】本実施形態のネットワークコンピュータ管理方法を実施する実施装置において、クライアント端末装置110がプログラムサーバ装置120に発行する要求のデータ構造を示す図である。

【図7】本実施形態のネットワークコンピュータ管理方法を実施する実施装置の起動接続プログラム700の処理手順を示すフローチャートである。

【図8】本実施形態のネットワークコンピュータ管理方法を実施する実施装置のネットワークプログラムロードプログラム800の処理手順を示すフローチャートである。

【図9】本実施形態のネットワークコンピュータ管理方法を実施する実施装置のマシン構成情報通知プログラム900の処理手順を示すフローチャートである。

【図10】本実施形態のネットワークコンピュータ管理方法を実施する実施装置のプログラム転送プログラム1000の処理手順を示すフローチャートである。

【図11】本実施形態のネットワークコンピュータ管理方法を実施する実施装置の転送アプリケーション選択プログラム1100の処理手順を示すフローチャートである。

【図12】図11の転送アプリケーション選択プログラム1100の処理途中において実行される、実行可能プ



ログラム抽出処理 1 2 0 0 の処理手順を示すフローチャートである。

【図 1 3】図 1 2 の実行可能なプログラムを抽出する処理 1 2 0 0 の処理における、格納する抽出結果テーブル 1 3 0 0 の一例を示す図である。

【図 1 4】図 1 1 の転送アプリケーション選択プログラム 1 1 0 0 における実行時優先度検査処理 1 4 0 0 の処理手順を示すフローチャートである。

【図 1 5】図 1 4 の実行時優先度検査処理 1 5 0 0 における優先度 n の属性について条件判定式を選択し、実行する処理 1 5 0 0 の処理手順を示すフローチャートである。

【図 1 6】図 1 5 に示した「判定式\_性能」、「判定式\_メモリ」、「判定式\_バージョン」の具体的な計算式を示す図である。

【図 1 7】図 1 4 の実行時優先度検査処理 1 4 0 0 の終了後における抽出結果テーブル 1 3 0 0 の一例を示す図である。

【図 1 8】図 1 1 の転送アプリケーション選択プログラム 1 1 0 0 における転送プログラム最終決定処理 1 8 0 0 の処理手順を示すフローチャートである。

【図 1 9】図 1 8 の転送プログラム最終決定処理 1 8 0 0 における選択例外処理 1 9 0 0 の処理手順を示すフローチャートである。

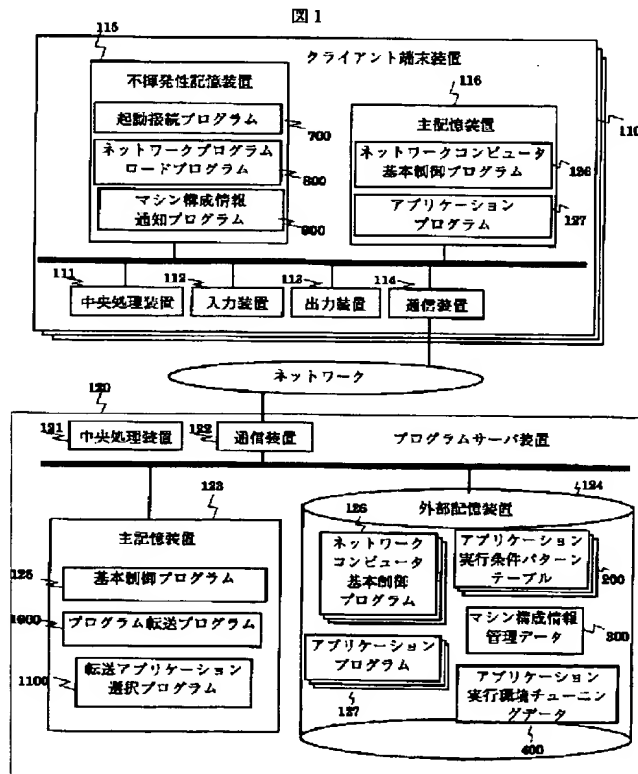
【図 2 0】図 1 1 の転送アプリケーション選択プログラム 1 1 0 0 における基本制御プログラム再送処理 2 0 0 0 の処理手順を示すフローチャートである。

【図 2 1】図 2 0 の基本制御プログラム再送処理 2 0 0 0 のステップ 2 0 0 4 において、クライアント端末に送信された基本制御プログラムの再送を促すメッセージの表示画面の一例を示す図である。

【符号の説明】

1 1 0 …クライアント端末装置、 1 2 0 …プログラムサーバ装置、 2 0 0 …アプリケーション実行条件パターンテーブル、 3 0 0 …マシン構成情報管理データ、 4 0 0 …アプリケーション実行環境チューニングデータ、 7 0 0 …起動接続プログラム、 8 0 0 …ネットワークプログラムロードプログラム、 9 0 0 …マシン構成情報通知プログラム、 1 0 0 0 …プログラム転送プログラム、 1 1 0 0 …転送アプリケーション選択プログラム。

【図 1】



【図 3】

301 ネットワークアドレス	302 基本制御プログラム	303 空きメモリ量 (KB)	304 CPU (MHz)
NCID00010	OS_A 1.0	250	180
NCID00012	OS_A 1.0	200	120
NCID00020	OS_B 1.0	100	140
NCID00022	OS_A 1.0	200	250
NCID00023	OS_B 1.1	300	220
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

【図 4】

401 ユーザ ID または セッション番号	402 アプリケーション実行環境チューニングパラメータ		
	403 性能	404 メモリ	405 バージョン
ritsuko	0.7	0.2	0.1
205	0.1	0.3	0.6
.	.	.	.
.	.	.	.
.	.	.	.

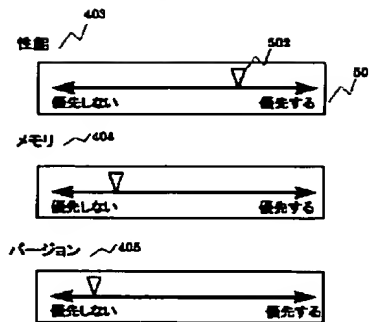
【図2】

図2

201	パターンID		1	2	3	4	5
202	アプリケーションプログラムA		Ver1.0	Ver1.1	Ver2.0	Ver3.0	Ver4.0
203	基本制御プログラム		OS_A1.0	OS_A1.0	OS_B1.0	OS_B1.0	OS_B1.1
204	必要メモリ量 (KB)	206 必須条件 207 推奨条件	100	200	300	400	500
205	CPU (MHz)	206 必須条件 207 推奨条件	100	150	200	250	300

【図5】

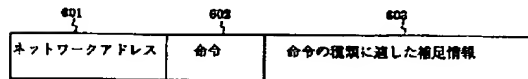
図5



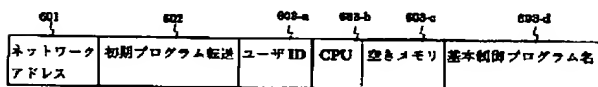
【図6】

図6

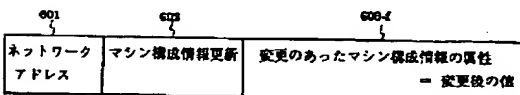
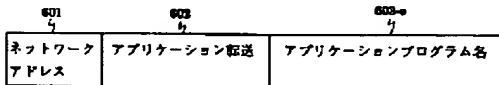
(A) 命令 = 「初期プログラム転送」の場合



(B) 命令 = 「アプリケーション転送」の場合

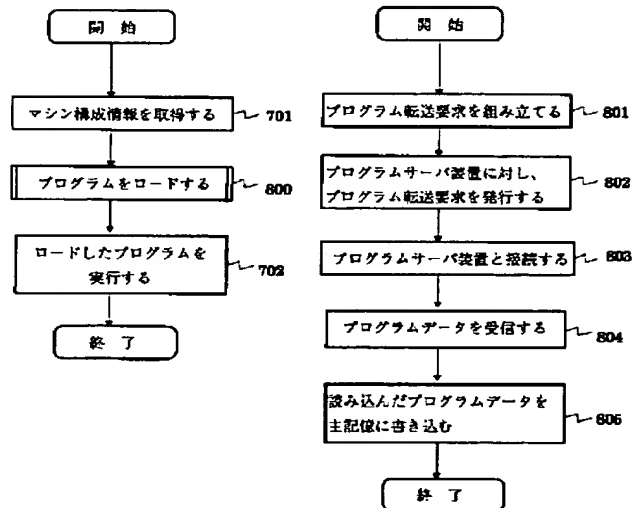


(C) 命令 = 「マシン構成情報更新」の場合



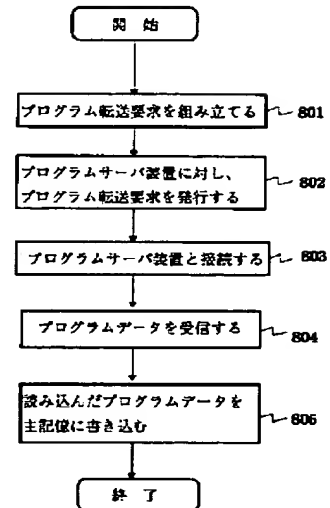
【図7】

図7



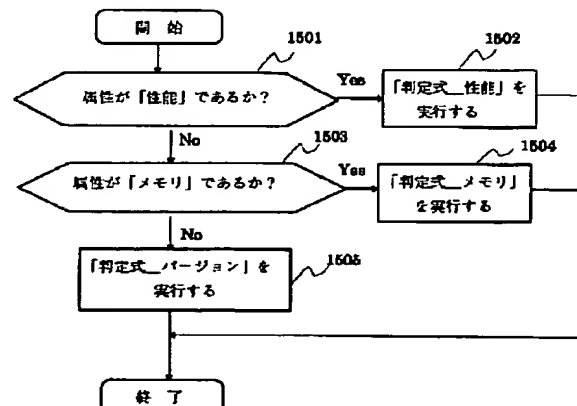
【図8】

図8

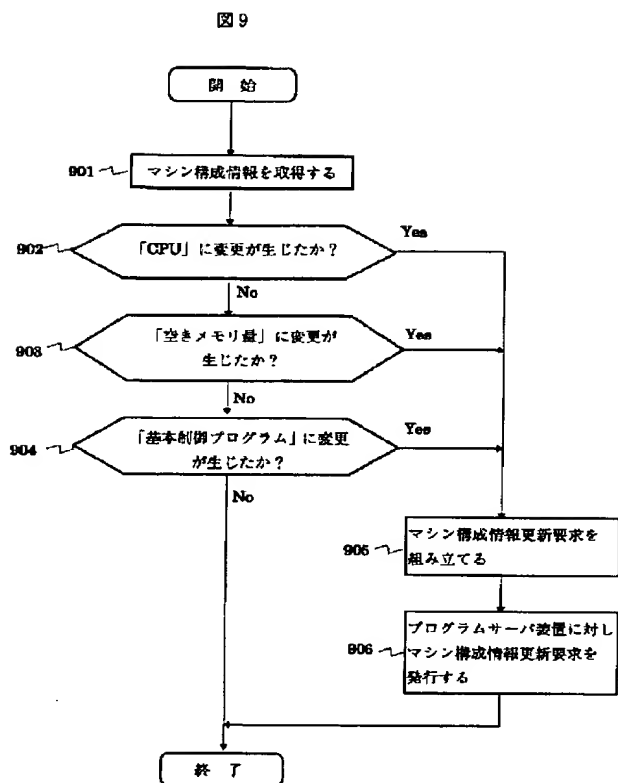


【図15】

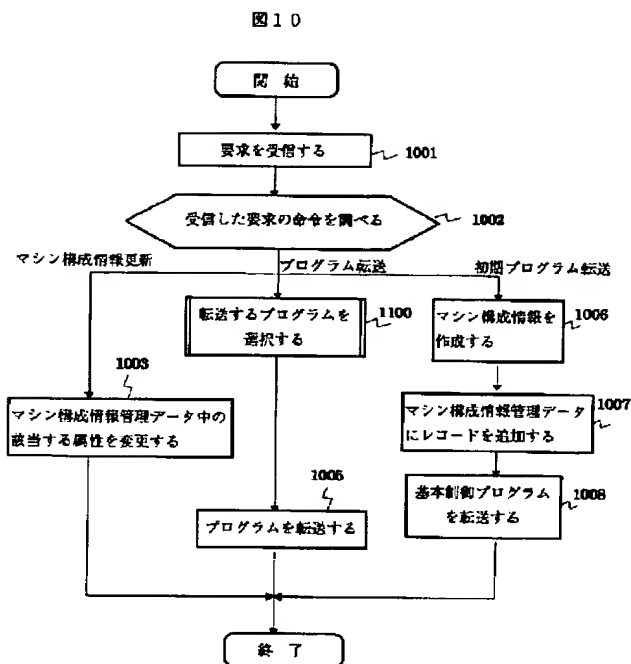
図15



【図9】

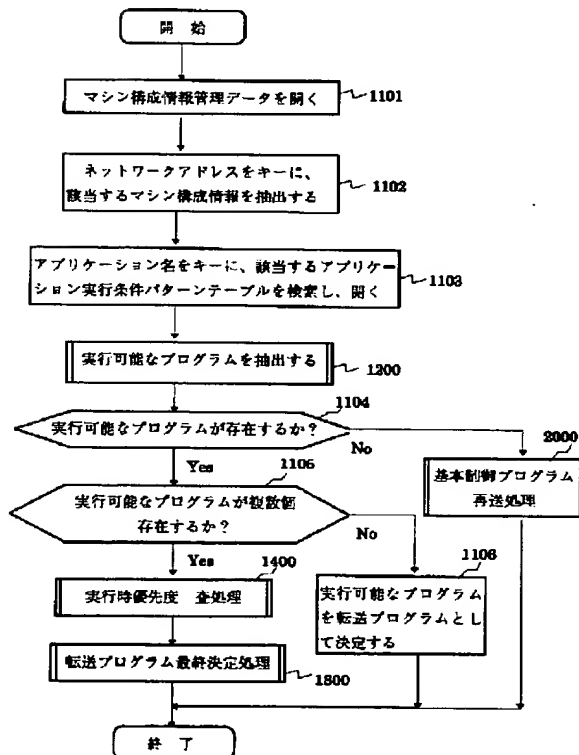


【図10】



【図11】

図11

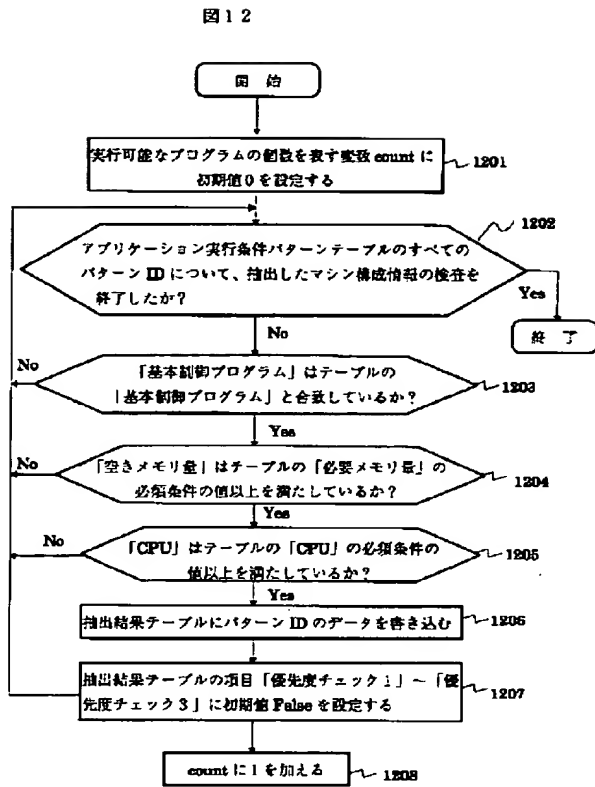


【図13】

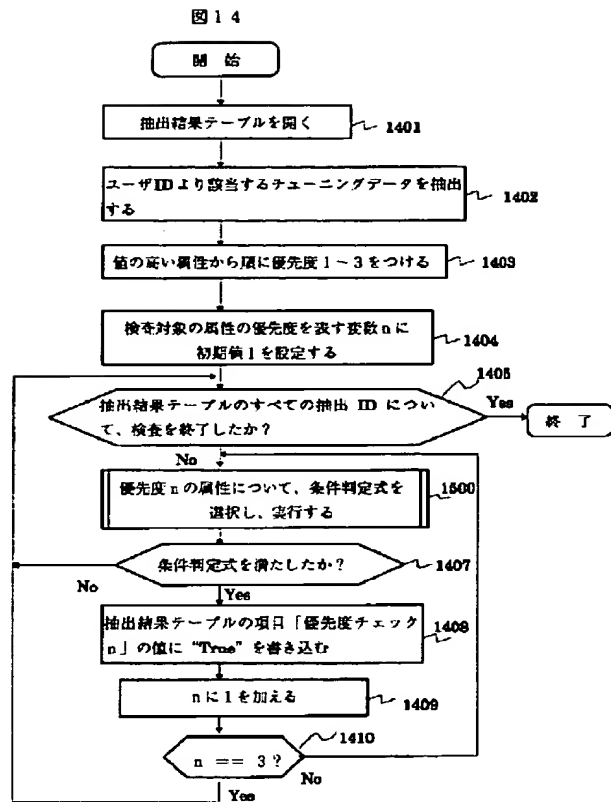
図13

1801	抽出 ID		1	2
201	パターン ID		1	2
202	アプリケーションプログラム A		Ver1.0	Ver1.1
203	基本制御プログラム		CS_A 1.0	CS_A 1.0
204	必要メモリ量 (KB)	必須条件	100	200
		推奨条件	200	300
205	CPU (MHz)	必須条件	100	150
		推奨条件	150	200
1302	優先度チェック 1		False	False
1303	優先度チェック 2		False	False
1304	優先度チェック 3		False	False

【図12】



【図14】



【図17】

図17

抽出 ID		1	2
パターン ID		1	2
アプリケーションプログラム A		Ver1.0	Ver1.1
基本制御プログラム		OS_A 1.0	OS_A 1.0
必要メモリ量 (KB)	必須条件	100	200
	推奨条件	200	300
CPU (MHz)	必須条件	100	150
	推奨条件	150	200
1701 優先度チェック 1		True	False
1702 優先度チェック 2		True	False
1703 優先度チェック 3		True	False

【図21】

図21

アプリケーションプログラム A を実行するには、  
基本制御プログラムをリロードする必要があります。  
リロードを希望しますか?

2101  
はい

2102  
いいえ

【図16】

図16

## ◆ 判定式\_性能

1801  

$$\begin{aligned} & (\text{アプリケーション実行条件パターンテーブルの「CPU」の必須条件の値}) + \\ & (\text{抽出したチューニングデータの「性能」の値}) \times \\ & | (\text{アプリケーション実行条件パターンテーブルの「CPU」の推奨条件の値}) \\ & - (\text{アプリケーション実行条件パターンテーブルの「CPU」の必須条件の値}) | \\ & \leq (\text{抽出したマシン情報の「CPU」の値}) \end{aligned}$$

## ◆ 判定式\_メモリ

1602  

$$\begin{aligned} & (\text{アプリケーション実行条件パターンテーブルの「必要メモリ量」の必須条件の値}) \\ & + (\text{抽出したチューニングデータの「メモリ」の値}) \times \\ & | (\text{アプリケーション実行条件パターンテーブルの「必要メモリ量」の \\ & 推奨条件の値}) - \\ & (\text{アプリケーション実行条件パターンテーブルの「必要メモリ量」の \\ & 必須条件の値}) | \\ & \leq (\text{抽出したマシン情報の「空きメモリ量」の値}) \end{aligned}$$

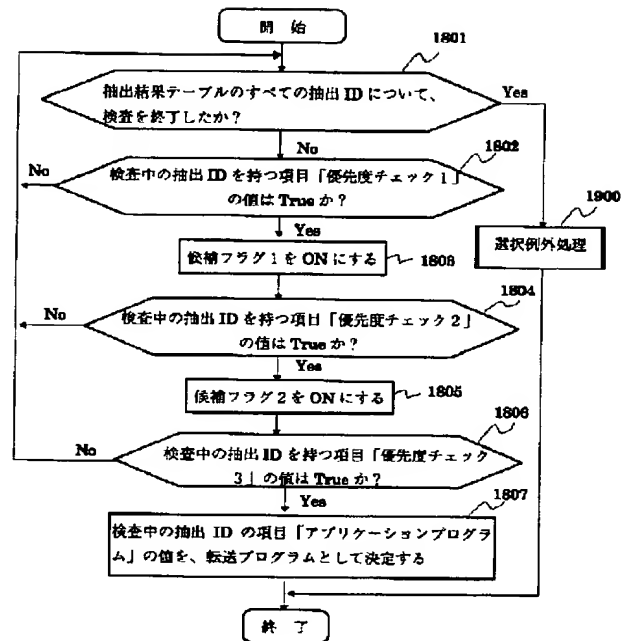
## ◆ 判定式\_バージョン

1603  

$$\begin{aligned} & 10 \times (\text{抽出したチューニングデータの「バージョン」の値}) \div \\ & (\text{実行可能なプログラムの個数}) \leq \text{検査対象としている抽出 ID} \\ & \leq 10 \times (\text{抽出したチューニングデータの「バージョン」の値}) \div \\ & (\text{実行可能なプログラムの個数}) + 1 \end{aligned}$$

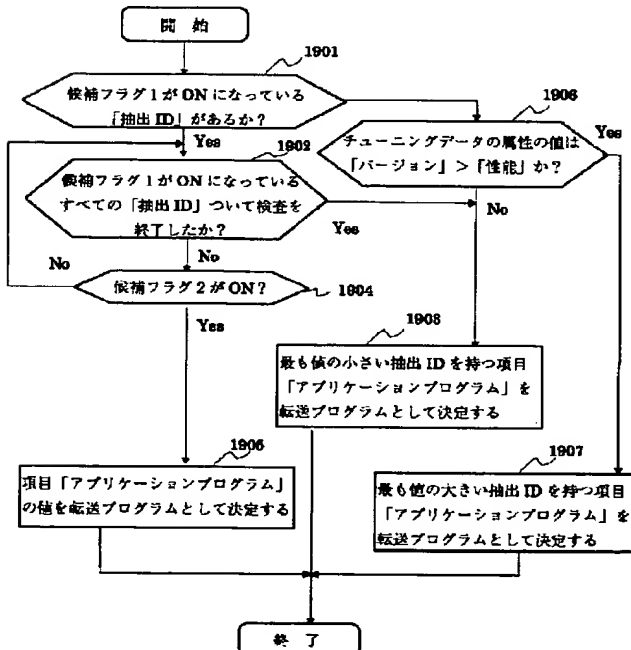
【図18】

図18

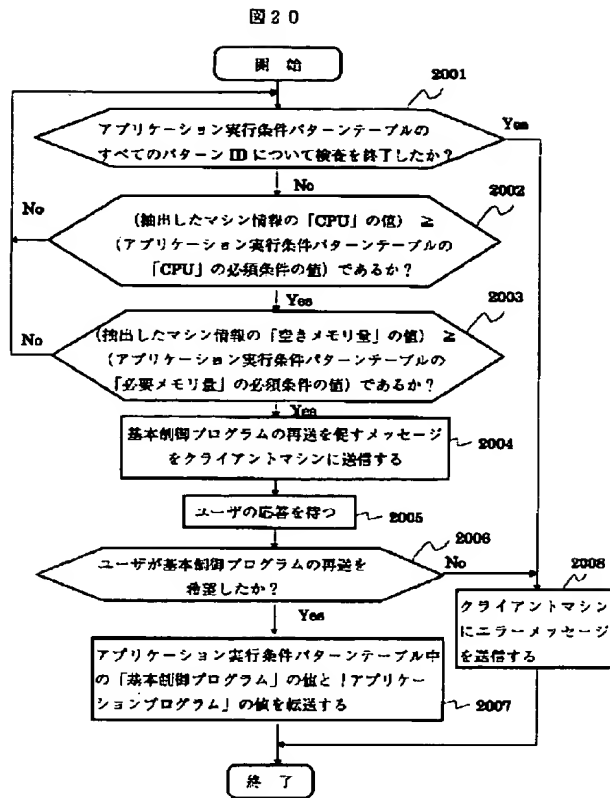


【図19】

図19



【図20】



**THIS PAGE BLANK (USPTO)**